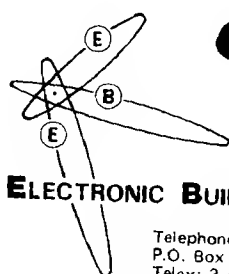


August 1979



Quantum
S.A. Distributor

ELECTRONIC BUILDING ELEMENTS (PTY) LTD
PURVEYORS OF ALL ELECTRONIC COMPONENTS
Telephone: 78-9221/6 Pine Square Berea Street
P.O. Box 4609, Pretoria (2nd Floor) Hazelwood
Telex: 3-0181 SA Pretoria

A High-Speed Emulator for Intel MCS-48TM Microcomputers

Applications Staff
Microcontroller Operation

Related Intel Publications

MCS-48 Microcomputer User's Manual, 9800270

MCS-48 Assembly Language Programming Manual, 9800255

The material in this Application Note is for informational purposes only and is subject to change without notice. Intel Corporation has made an effort to verify that the material in this document is correct. However, Intel Corporation does not assume any responsibility for errors that may appear in this document.

The following are trademarks of Intel Corporation and may be used only to describe Intel products: Intel, ICE, INSITE, Inteltec, iSBC, Library Manager, MCS, Megachassis, MICROMAP, MULTIBUS, PROMPT, RMX, UPI, μ Scope, PROMWARE, ICS and the combination of MCS, ICE, iSBC, ICS, RMX or HSE with a numerical suffix.

All mnemonics copyrighted©Intel Corporation 1976.

A High-Speed Emulator for Intel MCS-48™ Microcomputers

Contents

I. PURPOSE AND SCOPE.....	1
II. The HSE-49™ DEVELOPMENT TOOL.....	1
III. GENERAL HARDWARE OVERVIEW.....	1
IV. INTERPROCESSOR COMMUNICATION.....	3
V. COMMAND DESCRIPTION.....	4
VI. SYSTEM LIMITATIONS.....	7
VII. HARDWARE CONFIGURATIONS.....	8
APPENDIX A. SCHEMATIC DIAGRAMS.....	10
APPENDIX B. MONITOR LISTINGS.....	1
APPENDIX C. COMMAND SUMMARY.....	87
APPENDIX D. ERROR MESSAGES.....	87

I. PURPOSE AND SCOPE

This Application Note presents a description of the design and operation of a high-speed emulator for the Intel® MCS-48™ family of single chip microcomputers. The HSE-49™ emulator provides a simple and inexpensive means for executing and debugging 8049 programs which require the full 11-MHz operating speed of the part.

Section II of this Application Note describes some of the features of this development tool and how it may be used. Section III briefly discusses the hardware used to implement these features, while Section IV describes the manner in which program execution status is made available to the operator.

A detailed description of all of the operator commands is presented in Section V of this note, along with the modifiers and options which may be specified for each command. Known restrictions and limitations of the HSE-49 system are listed and explained in Section VI. Section VII shows how the basic circuit may be modified to provide options on memory organization, I/O configurations, etc.

Full schematics of the system hardware, as well as monitor software listings, are presented in Appendices A and B, respectively. A short summary of the command syntax is presented in Appendix C. Appendix D explains the error message codes which may appear during use.

It is assumed that the reader is already familiar with the operation of the 8048 or 8049 microcomputers. Some knowledge of the 8048 architecture is needed to understand sections of the command and modifier descriptions. Most users will already have this background. Other readers are referred to the *MCS-48 Microcomputer User's Manual*, Intel publication number 9800270.

II. THE HSE-49 DEVELOPMENT TOOL

In essence, the HSE-49 emulator provides the user a means for executing an MCS-48 program located in external RAM rather than internal ROM or EPROM. This allows programs being debugged to be modified easily and quickly during the debug cycle. A user's program may be entered into system RAM either manually or via a serial link from a host computer such as an Intellec® Microcomputer Development System. Once loaded, the program can be modified using an on-board keyboard and display, and executed in real-time in a number of breakpoint modes. The internal state of the processor, including RAM, accumulator, timer/counter, and status register contents, can also be read and modified through the keyboard.

Breakpoint and debug facilities are extremely flexible. The following execution modes are provided.

- Programs may be run in full (11 MHz) real time;
- Programs may be single-stepped;
- In break mode, programs run in full real time until break occurs;

- Breaks may be triggered by either program or external data RAM accesses;
- Any number of breakpoints may be used in any combination;
- "Auto-Step" operation causes the current program counter and Accumulator contents to be printed on the display for a short time on every instruction cycle;
- "Auto-Break" provides the above display only when a break flag is encountered, with real time operation otherwise;
- While running in non-break mode, a TTL-level pulse is generated whenever a break flag is encountered. This signal may be used to trigger an oscilloscope or Logic Analyzer to assist in hardware and software debug.
- While running in any mode, the keyboard and display are "alive". Execution may be suspended or terminated by commands from the keyboard.

Intent of this Note

While the HSE-49 emulator can assist a new microcomputer user in becoming familiar with the 8048 and 8049 microcomputers, its inherent debug capabilities will also prove helpful to design engineers. The design could be used for new system development and verification or adapted for prototype production.

The main concern in designing the HSE-49 emulator was to keep the basic design simple, while maximizing the system's flexibility. The design allows the use of jumpers, hardware and software switches, etc. to allow the user to reconfigure the system according to the way he dedicates chip-select pins, I/O, etc. The emulator can be changed to fit each user's unique needs, rather than forcing the user to alter his needs to what is provided.

The primary intent of note is to provide the reader with the information needed to reconstruct and make full use of the HSE-49 emulator. Less emphasis is placed on describing how the hardware operates or how the commands are implemented. This information may be found in the schematic diagrams and software listings included in the Appendices.

III. GENERAL HARDWARE OVERVIEW

User Program Emulation

The actual emulation of the user's program is done using an 8039 microcomputer (IC29 on the schematics in Appendix A) executing a program stored in external RAM. The basic minimum configuration includes the 8039 microcomputer, an 8282 address latch (IC19), and 2K bytes of 2114 RAM to use for program development and real-time execution (ICs B1, C1, B2, and C2). Additional RAM may be added to allow the user to expand his program and data memory to 4K each. (If an 11-MHz crystal is used with the microcomputer, type 2114-3 RAMs must be used.)

System Supervision

A second microcomputer — another 8039 (IC25) with an 8282 address latch (IC16) and off-chip program memory in a 2716 EPROM (IC15) — is used to scan the on-board keyboard and display, interpret and implement commands, drive serial interfaces, etc. In general, the master processor is used to interface the execution processor's memory spaces with the outside world and control the operation of the execution processor. In this note the two processors will be abbreviated "MP" and "EP", respectively. Figure 1 shows how the two processors interrelate with the rest of the system.

Keyboard/Display

The 33-key keyboard shown in Figure 2 includes a 16-key hexadecimal keypad and 17 special function keys for specifying commands and modifiers. Readers already

familiar with the PROMPT-48™ debug tool for the 8048 will find that 25 of the HSE-49 emulator keys are identical in function and layout to the PROMPT-48 keyboard, and use the PROMPT-48 command syntax. The eight additional keys are used to generalize and augment the PROMPT-48 capabilities, as described in Section V.

The eight-character seven-segment display (DS1-DS8) is used for displaying addresses, data, and pseudo-alphanumeric messages. The display responses printed in Section V and throughout this note use a mix of upper and lower case letters to indicate what seven-segment patterns appear. An 8243 (IC9) and eight DIP packages (resistor packs, current buffers, etc.) are used for multiplexing the display and scanning the keyboard.

Breakpoint Detection

Breakpoints are specified and detected using a 2102A 1K × 8 RAM corresponding to each pair of 2114s (ICs A1

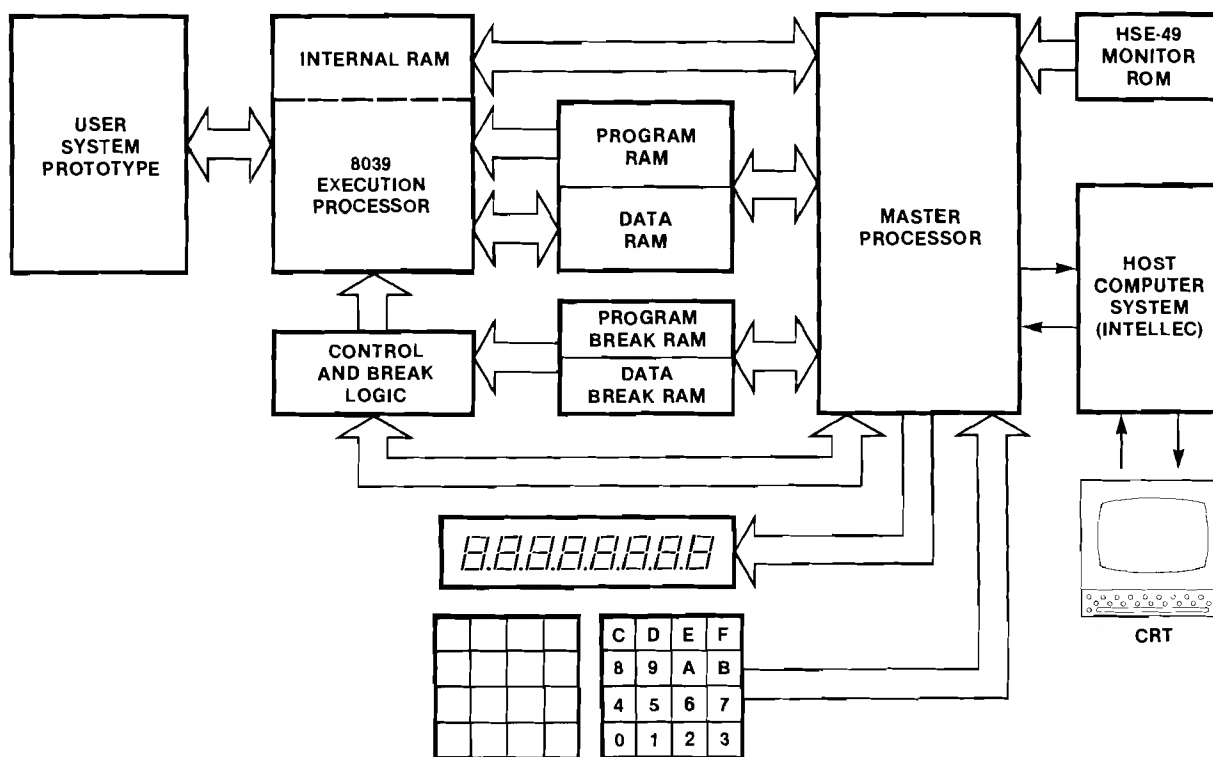


Figure 1. HSE-49™ Emulator Signal Flow Diagram

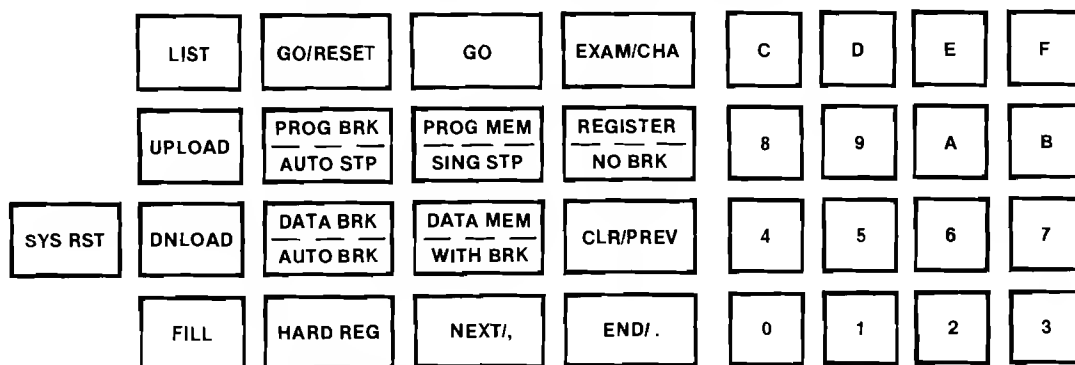


Figure 2. HSE-49™ Emulator Command Keyboard Organization

and A2). In effect, each program or data address accesses a 9-bit word. Eight bits are used normally for code or data storage. The ninth bit, accessed in parallel with the other eight, is used to indicate if a breakpoint has been set for that address. This output, when asserted, is latched (IC27 and IC36) and used to halt the execution processor via the single-step input. (In other modes, the break logic can be reconfigured to set the break requested flip-flop on any EP machine cycle or any EP "MOVX" instruction.)

Link Register

An 8212 8-bit latch (IC18) is used to communicate data and commands between the master and control processors. Under control of the MP, this register, called the "Link" register, may be logically mapped into either the program or data RAM address spaces. When this is done, the 2114s in the respective memory space are disabled and the link responds to all accesses, regardless of address. The link will be discussed in greater detail in Section IV.

Control Logic

In addition to the devices mentioned above, the minimum configuration requires about 10 additional ICs for bus arbitration, system control, and breakpoint and single-step logic. Additional parts may be optionally added for serial port interfacing, I/O reconstruction, etc.

MP Monitor

The monitor program executed by the MP includes commands for filling, reading, or writing the various memory spaces, including the execution processor's program RAM, external ("MOVX") data RAM, accumulator, PSW, PC, timer/counter, working registers, and internal RAM; to execute the user's program from arbitrary addresses in various debugging modes; and to upload or download object or data files from diskettes using an Inteltec® development system. No special software is needed for the Inteltec® other than ISIS Version 3.4 or later. The data format is compatible with the standard Intel hex file format produced by ASM-4; the baud rate may be altered from 110 baud (default state) up to 2400

baud from the on-board keypad. Blocks of data may be transmitted to a CRT or printer and displayed in a tabular format.

IV. INTERPROCESSOR COMMUNICATION

Program Break Sequence

When the MP detects that the EP has been halted by the breakpoint hardware, or when the operator presses a key while the program is executing, the program break sequence is initiated. The low-order 23 bytes of user program memory is read into a buffer within the internal RAM of the MP. A short program for reading and transmitting internal EP status is written over the low-order program memory. (This is one of several "mini-monitors" overlayed over the user program area.) The link register is mapped logically over the user program memory, and loaded with the 8049 machine code for a "CALL" instruction to the mini-monitor program area. The EP is then allowed to fetch a single instruction from the link, i.e., the "CALL" to the mini-monitor is forced onto the EP data bus.

From this point on, the EP executes code contained in the mini-monitor. The link is logically mapped over the data RAM address space (whether or not any 2114 data RAMs are present). A block diagram of the system at this point is shown in Figure 3. The break logic is reconfigured so that any "MOVX" (RD or WR) operation executed by the EP will cause it to halt.

For example, after entering the first mini-monitor, the EP executes a "MOVX @R0,A" instruction. This writes the contents of the accumulator prior to the execution termination into the link, and causes the EP to halt. The MP may then read and retain the link contents to determine the EP accumulator value. The EP timer/counter and PSW are preserved in the same manner.

Accessing EP Internal RAM

After reading and saving EP internal status, the MP loads a different mini-monitor into the same RAM area. This monitor allows the internal RAM of the EP to be read and written by the MP by passing address and data

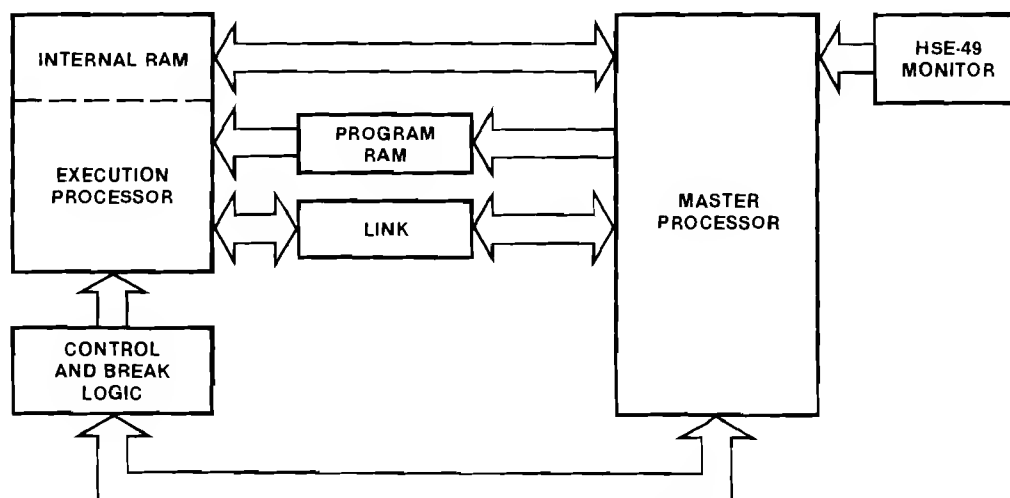


Figure 3. Communication between EP & MP

values between the two processors using the link register.

This is needed for two reasons. First, the EP program counter prior to the forced "CALL" instruction may be derived from the EP stack contents, and may be modified to cause the EP to resume execution at any desired address. Secondly, the internal RAM of the EP may then be accessed and modified in the process of executing a number of the monitor commands.

Resuming User Program Execution

In order to resume user program execution, a status-restoration mini-monitor is overlaid. This restores the EP internal status using a scheme analogous to the one in which the status was originally saved. The final step of the last mini-monitor is an "RETR" instruction, after which the EP is again halted. The low-order program memory saved earlier is rewritten into the appropriate area, the break logic is reconfigured for the desired execution mode, and the EP is released to run at full speed until the next break situation is encountered.

Note that all commands are implemented using "logical" rather than "physical" addressing. Thus the operator need not be concerned with the intricacies of the system design. For example, when any monitor command refers to low-order user program memory, the appropriate byte of storage within the MP internal RAM is accessed instead. If the location is altered, the internal RAM is modified appropriately. When program memory is reloaded prior to resuming user program execution, the modified version of the user program will be the one loaded.

Baud	HR06	HR07
110	93H	04H
150	96H	03H
300	45H	02H
600	9DH	01H
1200	44H	01H
2400	1AH	01H

Table 1. Serial Interface Data Rate Parameters

V. HSE-49 COMMAND DESCRIPTION

Whenever the characters "HSE-49" are present on the system display, a command string may be entered by the operator. In general, all command strings consist of a basic command initiator, an optional command modifier or type-designator, and a number of parameters or delimiters entered as hexadecimal digits. A command is executed, or a command in progress terminated, by pressing the [END/.] key. Logical default values are assumed for the modifier and parameters if either (or both) are omitted. A default parameter assumed for the command modifier will be presented on the display when the first parameter is entered.

Each parameter is a string of up to three hexadecimal digits. If more than three digits are entered, only the most recent three are considered. This allows an erroneous digit to be corrected without respecifying the entire command. A parameter is completed by pressing the [NEXT/.] key. Some commands may only need the

low order part of a parameter; i.e., a command incorporating a data byte (such as [FILL]) will use only the low-order 8 bits of the corresponding parameter; Internal RAM and hardware register addressing uses only seven. In each case, higher order bits are ignored.

A command string is terminated and the command invoked by pressing the [END/.] key. The command will also be invoked by pressing the [NEXT/.] key when no additional parameters are allowed. A command string may be aborted at any point before the command is invoked by pressing the [CLEAR/PREV] key, and the sign-on message will appear.

Errors

An illegal command string, command terminator, or hardware failure will cause an error message and error code number to appear on the display (e.g., "Error-.3"). When this occurs, the monitor can be returned to command mode by pressing the [CLEAR] or [END/.] keys. An explanation of the various error codes is given in Appendix D.

Command Classes

Commands for the HSE-49 emulator are divided into general classes, where all commands in each class have the same choice of options or modifiers. A brief description of each command, followed by a description of the allowed options, is presented below by class.

Data Manipulation/Control Command Group

Commands:

[EXAM/CHA]

Display Response — "ECh."

Function — Examine/change memory location.

Causes the memory address specified to be read and presented on the display. New data may be entered (if desired) from the hexadecimal keypad. New data is verified before appearing on the display. Subsequent or previous locations may be read by pressing the [NEXT/.] or [PREV] keys, respectively. Command terminated with [END/.] key.

[FILL]

Display Response — "FIL."

Function — Fill range of memory addresses with a single data value.

Fill the appropriate memory space between the addresses specified by the first two parameters with the low-order byte of the third parameter. If second parameter less than first, only the location specified by the first is affected. If third parameter omitted, zero is assumed. If second and third parameters omitted, individual address specified is cleared. Command is useful for setting a large range of breakpoints; e.g., all of page 3 may be enabled for break with the command:

[FILL][PROG BRK]<300>[,<3FF>[,<1>[.]

[LIST]

Display Response — "LSt."

Function — List memory to output device through HSE-49 serial port.

Display the contents of a range of addresses given by two parameters to a teletype or CRT screen. Data is formatted, 16 separated bytes per line, with the starting address of each line printed. If used with an Intellec® system, the operator first uses ISIS-II to transfer the TTY input to the CRT output ("COPY :Tl: TO :CO:") then invokes this command from the keypad. Alternatively, any ISIS device or disk file name(:TO:, :LP:, :F1:HRDREG.SAV, etc.) may be used as the destination.

[DNLOAD]

Display Response — "dnL."

Function — Download memory through HSE-49 serial port

Load data in hex file format through the serial input port. If used with Intellec® system, the operator first invokes this command from the keypad, then uses ISIS-II to transfer a disk file to the teletype port ("COPY :Fn:file.HEX TO :TO:").

The use of the checksum field for the download command is expanded slightly over the Intel hex file format standard. If the first character of the checksum field is a question mark ("?"), the checksum for that record will not be verified. This allows large object files produced by the assembler to be patched using the ISIS text editor without the necessity of manually recomputing the checksum value.

[UPLOAD]

Display Response — "UPL."

Function — Upload memory through HSE-49 serial port.

Output the contents of a range of addresses specified by the two parameters through the HSE-49 serial port in standard Intel hex file format. If used with Intellec® system, the operator first uses ISIS-II to transfer the TTY input to a disk file ("COPY :Tl: TO :Fn:file.HEX"), then invokes this command from the keypad.

Data types allowed:

[PROG MEM]

Display Response — "Pr."

Function — User program memory.

Memory used to develop and execute user program. Addresses 000 through 7FF are the execution processor's memory bank 0; 800 through FFF are memory bank 1.

[REGISTER]

Display Response — "rG."

Function — Register memory and RAM.

Internal RAM of execution processor. Locations 0-7 are working register bank 0; 18-1F are working register bank 1. Only the low-order 7 bits of an address are considered.

[DATA MEM]

Display Response — "dA."

Function — External data memory (if installed).

Memory accessed by execution processor "MOVX A,@Rr" or "MOVX @Rr,A" instructions. High-order 4 bits may or may not be relevant, depending on jumpering option selected (explained in Section VII of this note).

[HARD REG]

Display Response — "Hr."

Function — Hardware registers.

The execution processor (EP) hardware registers (accumulator, timer/counter, etc.), as well as several parameters for controlling HSE-49 system status, are accessible through this catch-all memory space. Addresses are as follows:

00 — EP accumulator.

01 — EP PSW.

Bits correspond to 8049 PSW except that bit 3 (unused in the 8049) is used to monitor and alter the state of F1. Bits 2-0 correspond to the stack pointer value after the EP executes a CALL to the mini-monitor; i.e., one greater than when EP was running the user's program.

02 — EP timer/counter.

03 — EP internal RAM location 00.

(This value is also accessible through [REGISTER] space.)

04 — EP program counter (low byte).

05 — EP program counter (high nibble).

06-07 — HSE-49 serial interface baud rate parameters. Defaults to 110 baud; other rates may be selected by loading the values listed in Table 1.

08 — HSE-49 automatic sequencing rate parameter. Used in [GO][AUTO STP] and [GO][AUTO BRK] execution commands. 00 → fastest; FF → slowest. Defaults to 20H; approximately two steps per second.

09 — Monitor version/release number (packed BCD).

0A-0F — Currently unused by the monitor program.

10-7F — Variables used by master processor (MP) monitor. Should not be altered by operator.

[PROG BRK]

Display Response — "Pb."

Function — User program breakpoint memory.

Memory space used to indicate points where program execution should halt when running in a mode with breakpoints enabled ([GO][W/ BRK] and [GO][AUTOBRK]). Break will occur if enabled byte is read as the first or last byte of a 2-byte instruction, or read in executing a MOVP, MOVP3, or JMPP instruction. Memory is only 1 bit per location; 00 indicates continue, 01 causes a halt. Addresses 000 through 7FF are the execution processor's memory bank 0; 800 through FFF are memory bank 1.

[DATA BRK]

Display Response — "db."

Function — External data RAM breakpoint memory.

Memory space used to indicate points where data accesses should halt when running in a mode with breakpoints enabled ([GO][W/ BRK] and [GO][AUTOBRK]). Memory is only 1 bit per location; 00 indicates continue, 01 causes a halt. High-order 4 bits of breakpoint address may or may not be relevant, dependent on jumpering option selected for the corresponding data RAM (explained in Section VII of this note).

User Program Execution Control Group

Commands:

[GO]

Display Response — "Go."

Function — Begin execution.

If a parameter is given as part of the command string, execution will begin at that address. Otherwise, the EP program counter (hardware registers 04 and 05) will be used. These will contain the program counter from an earlier program execution break unless they have since been explicitly modified by the operator.

If command is terminated by [END/.], the EP's F1, PSW and stack pointer will be cleared. If command string is terminated by [NEXT/.], PSW will be taken from the EP PSW contents (hardware register 01).

While running the user's program, the characters "-run-" are written on the display. Execution may be halted and another command initiated by pressing the appropriate command key. Execution may be suspended at any time in any mode by pressing the [END/.] key. This will cause the current value of the execution processor program counter and accumulator to appear on the display in the form "PC.234-56". System status is saved in the appropriate hardware registers. At this point, or when an enabled breakpoint is encountered, pressing the [NEXT/.] key will cause the program to continue in the same mode as before. Any other command may be invoked by pressing the appropriate command string.

[GO/RESET]

Display Response — "Gr."

Function — Go from reset state.

EP is hardware-reset and released to execute the user's program from location 000H. No parameters are allowed. F0, F1, PSW, stack pointer, memory bank flip-flop, etc., are cleared.

Note that this command does not require the use of mini-monitors to initiate program execution. As the last phase of the program development cycle, the 2114 program RAMs and address decoder may be removed and replaced by a ROM or EPROM part (not shown in schematics). This command may be used to start execution when the program RAM has been removed. No interrogation of EP status or internal RAM may be done, nor are break or single-step modes allowed in this case, though the 2102A breakpoint RAM outputs may still be used to trigger a logic analyzer.

Execution modes allowed:

[NO BRK]

Display Response — "nb."

Function — Without breakpoints.

Full-speed execution without breakpoints enabled. Does not affect the state of the breakpoint memories.

[SING STP]

Display Response — "SSt."

Function — Single Step.

Step through program one instruction at a time. After each instruction is executed, execution halts with the current value of the Execution Processor Program Counter and Accumulator appearing on the display in the form "PC.234-56". System status is saved in the appropriate Hardware Registers. At the point, [NEXT/.] will cause the program to execute one more instruction, or any other command may be invoked by pressing the appropriate command string. Does not affect the state of the Breakpoint Memories.

[W/ BRK]

Display Response — "br."

Function — With breakpoints.

Full-speed execution with breakpoints enabled. When a breakpoint is encountered, execution halts with the current value of the execution processor program counter and accumulator appearing on the display in the form "PC.234-56". System status is saved in the appropriate hardware registers. At this point, [NEXT/.] will cause the program to continue until the next breakpoint is reached, or any other command may be invoked by pressing the appropriate command string.

[AUTO STP]

Display Response — "ASSt."

Function — Automatically sequence through a series of instructions.

Step through program one instruction at a time. After each instruction is executed, execution halts with the current value of the execution processor program counter and accumulator appearing on the display in the form "PC.234-56". System status is saved in the appropriate hardware registers. Execution resumes after a time determined by contents of hardware register 08. Does not affect the state of the breakpoint memories.

[AUTO BRK]

Display Response — "Abr."

Function — Automatically sequence between breakpoints.

Execute a series of instructions in real time between breakpoints. When breakpoint is encountered, halt EP temporarily while program counter and accumulator contents are displayed, then continue. Display is sustained after execution resumes. Does not affect the state of the breakpoint memories.

Breakpoint Control Command Group

Commands:

[B]

Display Response — "Stb."

Function — Breakpoint set.

Set breakpoint for the address given. Multiple breakpoints may be set by entering additional addresses, separated by the [NEXT/,] key. Command terminated by pressing [END/]. Action taken is to fill the appropriate breakpoint memory locations with logical ones.

[C]

Display Response — "CLb."

Function — Clear breakpoint.

Clear breakpoint for the address given. Multiple breakpoints may be cleared by entering additional addresses, separated by the [NEXT/,] key. Command terminated by pressing [END/]. Action taken is to fill the appropriate breakpoint memory locations with logical zeroes.

Data types allowed:

[PROG MEM]

Display Response — "Pr."

Function — Break on program memory fetch.

Applies command to the program breakpoint memory space.

[DATA MEM]

Display Response — "dA."

Function — Break on data memory access.

Applies command to the external data breakpoint memory space.

System Control Command Group

Command:

[SYS RST]

Display Response — "HSE-49."

Function — System reset.

Reset both the MP and EP and clear all breakpoints (requires approximately one second). CAUTION — If reset while EP is executing the user's program, the low order section of program memory (about 23 bytes) will be altered.

VI. SYSTEM LIMITATIONS

In designing the HSE-49 emulator, certain compromises were made in an attempt to maximize the usefulness of the emulator while keeping the circuitry simple and inexpensive. As a result, the following limitations exist and must be taken into account when using the system.

1. As explained in Section IV, user program execution is terminated (by single-stepping, breakpoints, pressing the [END/.] key, etc.) by forcing the execution processor to execute a "CALL" instruction to the mini-monitor. This uses one level of the EP subroutine stack. The EP PSW reflects the value of the stack pointer *after* processing this CALL. As a result, the value indicated for stack depth by examining the EP PSW (hardware register 01) is one greater than the depth when the break was initiated. The user program must not be using all eight levels of stack when a break is initiated or the bottom level will be destroyed.
2. User program is initiated (by the [GO] command or when resuming execution after a breakpoint, single-stepping, etc.) by forcing the EP to execute an "RETR" instruction. This will clear the EP interrupt-in-progress flip-flop. If the user program allows both external and timer interrupts to be enabled at the same time, care must be taken to avoid causing a break while the EP is within an interrupt servicing routine. No limitation is placed on breakpoints or single-stepping in the background program because of this.
3. When the user program execution is terminated (by a break, single-stepping, etc.) and later resumed, the EP timer/counter is restored to its value when the break occurred (unless modified by the user). The prescaler, however, will have changed. Thus, up to 31 machine cycles may be "lost" or "gained" if a break occurs while the timer is running.
4. Timer interrupts occurring at the same time as an EP break may be ignored if the timer overflow occurs after breaking user program execution before the timer value is saved.
5. The 8049 "RET" and "RETR" instructions are each 1-byte, 2-cycle instructions. During the second cycle the byte following the return instruction is fetched and ignored. If a program breakpoint is set for a location following a "RET" or "RETR" instruction, a break will be initiated when the return is executed.

6. Breakpoints should not be placed in the last 3 bytes of an EP memory bank (locations 7FDH-7FFH and 0FFDH-0FFFH). User program should not be single-stepped or auto-stepped through these locations.
7. Since I/O configuration is determined by external hardware rather than software, I/O modes may not be altered while a program is executing. (See Section VII for further details.)
8. The "ANL BUS,#nn" and "ORL BUS,#nn" instructions may not be used in the user program, as external hardware cannot properly restore these functions.
9. The memory bank select flag is not affected by the user program break sequence. Upon resuming execution with the [GO] command this flag will remain in the same state as before the preceding break. The flag may be cleared only by executing the [GO/RESET] or [SYS RST] commands.

VII. HARDWARE CONFIGURATIONS

A number of control and status lines are available to the user. All are low-power Schottky TTL-compatible signals.

TP1 — Unused MP input.

TP2 — Unused MP output.

TP3 — User program suspended. Low when EP running user code. High when halted or running mini-monitors.

TP4 — Breakpoint encountered. Normally low. High-level pulse generated when breakpoint passed. Useful for triggering logic analyzers, oscilloscopes, etc.

TP5 & TP6 — Memory matrix mode control. Select program vs. data RAM, link mapping configuration, etc. (See Appendix B for details.)

TP7 — Bus control. Low when MP controls common memory buses. High when EP controls memory buses.

The HSE-49 emulator hardware is designed to allow the user to reconfigure the system for a wide variety of different applications by installing or removing jumper wires or additional components. The schematics in Appendix A show the components needed for a variety of different configurations. In general, not all of the devices are required (or allowed) for any one configuration. The devices which are required are included in the following description.

The types of options allowed are divided below into several general classes and subdivided into mutually-independent features. Within some of these features there are numbered, mutually exclusive configurations; i.e. the serial interface (if desired) may use either

current-loop or RS-232C current buffers, but not both at one time.

Standard Operating Configuration

(Minimum system configurations — up to 4K program RAM; no data RAM; no serial interfaces; no execution processor I/O reconstruction.)

A. Basic 2K monitor from Appendix B:

Install resistors R4-R6
 Install transistor Q1
 Install crystals Y1-Y2
 Install capacitors C5-C38
 Install switches S1-S33
 Install displays DS1-DS8
 Install IC1-IC2
 Install RP3-RP5
 Install IC6-IC7
 Install RP8
 Install IC9
 Install IC15-IC20
 Install IC25-IC30
 Install IC34
 Install IC36-IC38
 Install A1-A2
 Install B1-B2
 Install C1-C3
 Install jumpers 13-15
 Install jumpers 17-18
 Install jumper 20

B. Expansion 2K monitor:

Install IC14
 Remove jumper 17

Serial Interface Buffer Selection

A. Current loop serial interfaces (4N46s) installed for use with full Intellec® Model 800 development system TTY port.

Install IC21-IC22
 Install resistor R1-R3
 Install jumpers 4-9
 (Remove RS-232 jumpers)

B. RS-232C serial interfaces (MC1488 and MC1489) installed for use with CRT as output device for data dumps:

Install IC23-IC24
 Install jumpers 1-3
 Install jumpers 10-11
 (Remove current-loop jumpers)

External Data RAM Address Decoding Scheme for Execution Processor

A. Up to 16 pages of on-board external data RAM installed for execution processor (addresses 0 through

0FFFH = 4K bytes); port 2 used for addressing pages 0 through 15:

Install jumpers 21-25
Install jumper 27
Install A5-A8
Install B5-B8
Install C5-C8

- B. One page of on-board external data RAM installed for execution processor (addresses 0 through 0FFFH); port 2 not used for data addressing:

Install jumper 26
Install jumper 28
Install A5
Install B5
Install C5

Connect the outputs of IC20, pins 7, 9, 10, & 11 to the inputs of a 74LS21 AND gate (not shown). Connect the output to CE and CS inputs of A5-C5. (Note: these signals are all present at jumpers 21-24 on the schematics.)

Reconstructing I/O for Execution Processor

- A. Application of port 2, pins P23-P20:

- (1) Using P23-P20 for latched output data (used with "OUTL P2,A", "ANL P2,#data", and "ORL P2,#data" instructions):

Install IC31

- (2) Using P23-P20 for interfacing to an 8243 in user's prototype:

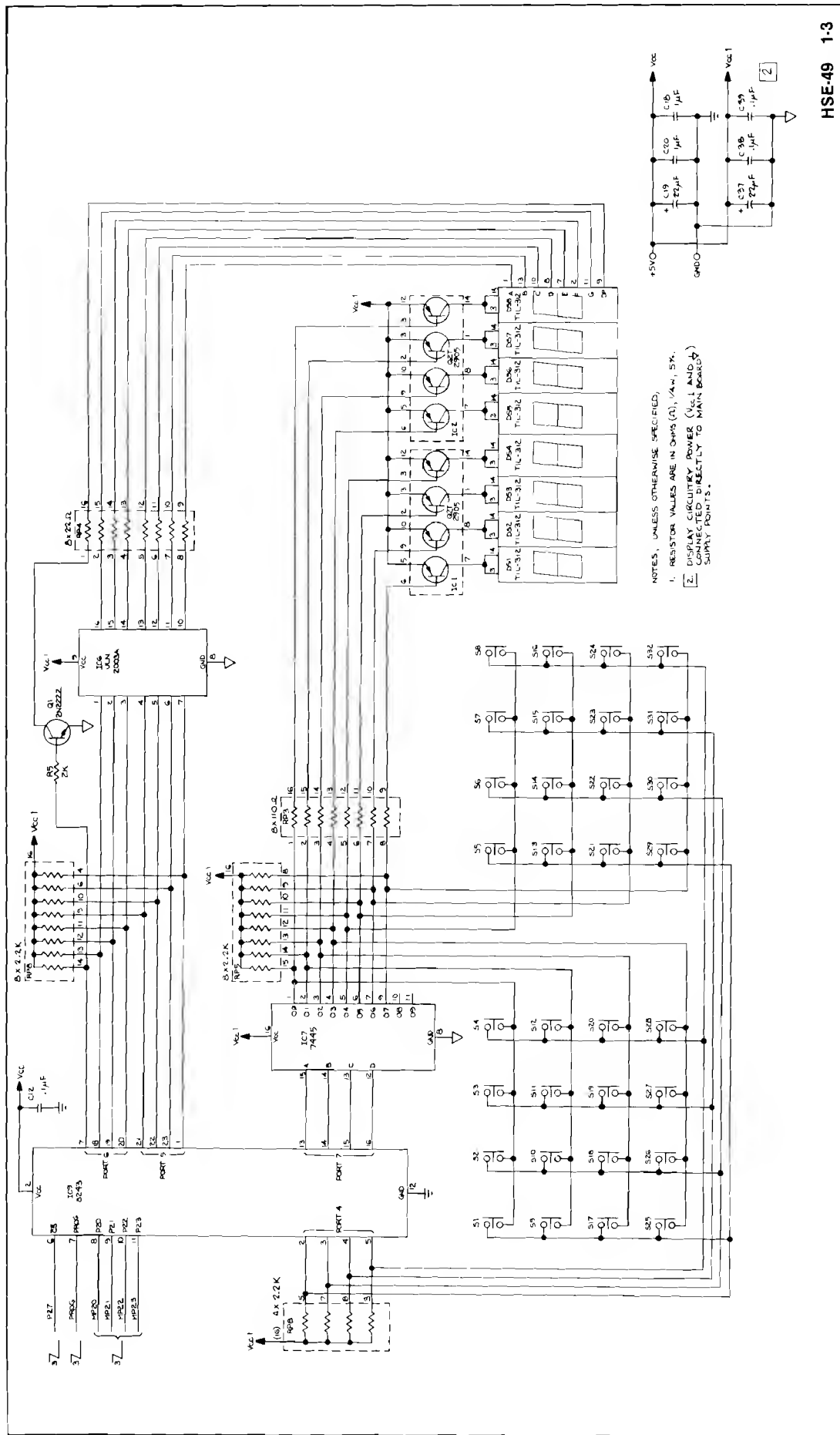
Connect D3-D0 pins on IC31 socket to corresponding Q3-Q0 pins.

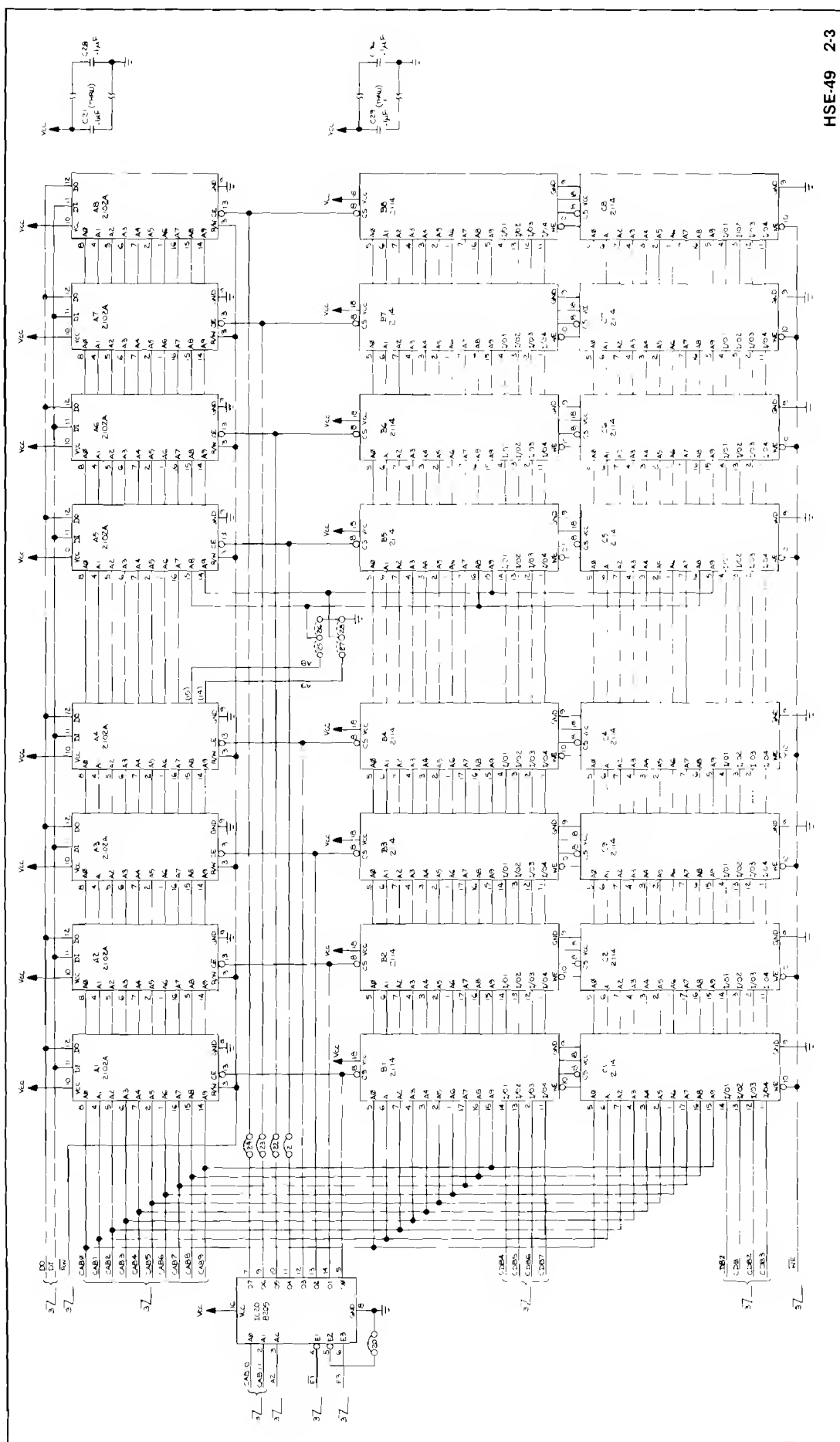
- B. Application of execution processor BUS:

- (1) Use of BUS as latched output port ("OUTL BUS,A"):

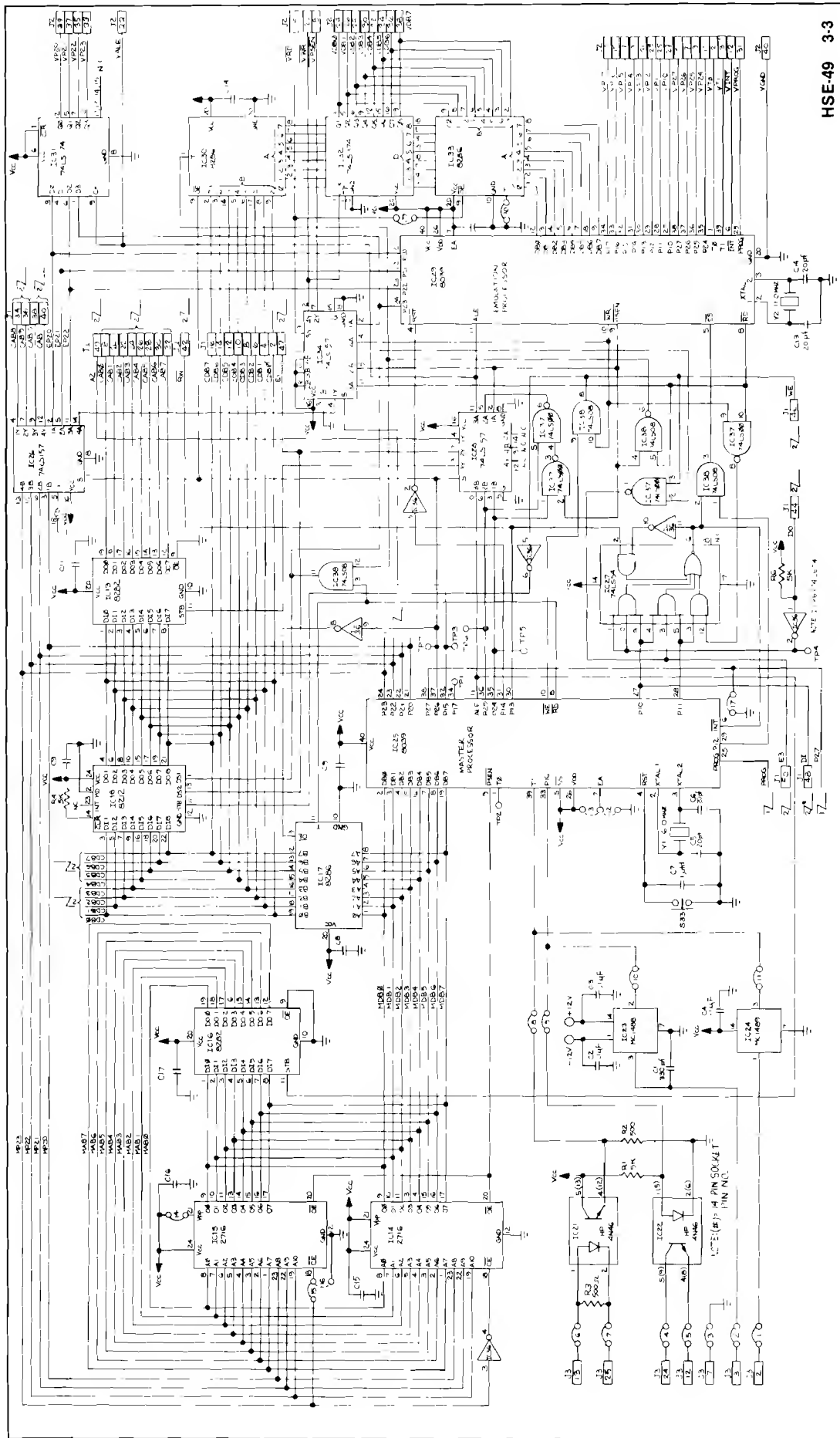
Install IC32

APPENDIX A
SCHEMATIC DIAGRAMS





HSE-49 2-3



APPENDIX B
MONITOR LISTINGS

ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0
HSE-49(TM) EMULATOR MONITOR VERSION 2.5

PAGE 1

LOC	OBJ	LINE	SOURCE STATEMENT
		1	\$MACROFILE NOGEN NOCOND XREF
		2	\$TITLE('HSE-49(TM) EMULATOR MONITOR VERSION 2.5')
		3	;
		4	*****
		5	;
		6	PROGRAM: HSE-49(TM) EMULATOR MONITOR
		7	VERS 2.5/789
		8	;
		9	COPYRIGHT (C) 1979
		10	INTEL CORPORATION
		11	3065 BOWERS AVENUE
		12	SANTA CLARA, CALIFORNIA 95051
		13	;
		14	*****
		15	;
		16	ABSTRACT
		17	=====
		18	;
		19	THIS PROGRAM CONTAINS THE SOFTWARE NECESSARY TO RUN THE HSE-49(TM)
		20	HIGH-SPEED EMULATOR FOR INTEL'S MCS-48(TM) FAMILY FAMILY OF MICROCOMPUTERS.
		21	THE EMULATOR PROVIDES AN ASSORTMENT OF UTILITY FUNCTIONS FOR
		22	DEVELOPING AND DEBUGGING 8049-BASED APPLICATIONS, INCLUDING THE
		23	ABILITY TO ENTER AND MODIFY PROGRAMS IN PROGRAM RAM,
		24	ALTER DATA, SINGLE-STEP SECTIONS OF A PROGRAM, AND EXECUTE PROGRAMS
		25	AT SPEEDS OF UP TO 11 MHZ, WITH OR WITHOUT BREAKPOINTS ENABLED.
		26	THE EMULATOR IS DESCRIBED IN GREATER DEPTH IN INTEL'S APPLICATION NOTE
		27	AP-55, "A HIGH-SPEED EMULATOR FOR INTEL MCS-48(TM) MICROCOMPUTERS."
		28	;
		29	PROGRAM ORGANIZATION
		30	=====
		31	;
		32	THIS LISTING IS ORGANIZED AS FOLLOWS:
		33	;
		34	INTRODUCTION AND HARDWARE OVERVIEW;
		35	VARIABLE DECLARATION AND DEFINITION;
		36	POWER-ON SYSTEM INITIALIZATION;
		37	KEYBOARD COMMAND PARSER AND ASSOCIATED TABLES;
		38	IMPLEMENTATIONS OF THE PRIMARY COMMANDS;
		39	DATA ACCESSING UTILITY SUBROUTINES USED THROUGHOUT;
		40	KEYBOARD SCANNING AND DISPLAY DRIVING SUBROUTINE;
		41	KEYBOARD AND DISPLAY INTERFACING UTILITIES;
		42	ROUTINES AND UTILITY SUBROUTINES WHICH INTERACT BETWEEN MP AND EP.
		43	;
		44	;
		45	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		46 ;	
		47 ;	INTRODUCTION AND HARDWARE OVERVIEW
		48 ;	=====
		49 ;	
		50 ;	THE EMULATOR DESIGN USES TWO MICROPROCESSORS. ONE PROCESSOR CONTROLS
		51 ;	SYSTEM STATUS, INTERPRETS MONITOR COMMANDS, AND COMMUNICATES
		52 ;	WITH THE OUTSIDE WORLD THROUGH THE ON-BOARD KEYBOARD, DISPLAY, SERIAL
		53 ;	INTERFACES, CONTROL SIGNALS, ETC.
		54 ;	A SECOND PROCESSOR IS USED TO ACTUALLY
		55 ;	EXECUTE THE USER'S PROGRAM UNDER THE CONTROL OF THE FIRST.
		56 ;	THESE PROCESSORS ARE REFERRED TO
		57 ;	THROUGHOUT THIS PROGRAM AS THE MASTER PROCESSOR (MP) AND EXECUTION
		58 ;	PROCESSOR (EP) RESPECTIVELY.
		59 ;	
		60 ;	THE PROGRAM IN THIS LISTING IS EXECUTED BY THE MASTER PROCESSOR.
		61 ;	AT THE END OF THIS LISTING ARE SEVERAL SHORT "MINI-MONITOR OVERLAYS"
		62 ;	WHICH THE EXECUTION PROCESSOR EXECUTES WHEN INTERACTION BETWEEN THE
		63 ;	TWO PROCESSORS IS NECESSARY.
		64 ;	
		65 ;	THIS PROGRAM WAS WRITTEN USING A NUMBER OF MACROS TO HANDLE THE ALLOCATION
		66 ;	OF MPU RESOURCES (WORKING REGISTERS, INTERNAL RAM, AND MP MONITOR ROM
		67 ;	FOR CODE AND DATA STORAGE). THESE MACRO DEFINITIONS ARE INCLUDED IN A FILE
		68 ;	NAMED "ALLOC.MAC," AND ARE PRINTED IN THIS LISTING FOR REFERENCE.
		69 ;	ANOTHER SET OF MACROS IS USED TO SIMPLIFY THE ACCESSING OF VARIABLES
		70 ;	STORED IN INTERNAL RAM (AS OPPOSED TO WORKING REGISTERS) BY USING R1 TO
		71 ;	INDIRECTLY ADDRESS THE APPROPRIATE RAM LOCATION WHEN NECESSARY.
		72 ;	THESE MACROS ARE INCLUDED IN "MOPCOD.MAC", AND ARE ALSO PRINTED HERE.
		73 ;	COMPLETE UNDERSTANDING OF THESE MACROS IS NOT REQUIRED TO UNDERSTAND THE
		74 ;	MONITOR PROPER; ALL LINES WHICH ACTUALLY PRODUCE OBJECT CODE APPEAR IN
		75 ;	THE LISTING ITSELF, INDENTED TWO SPACES FROM THE NORMAL TABULATION COLUMNS.
		76 ;	THE ACTUAL MONITOR PROGRAM FOR THE EMULATOR BEGINS AT APPROXIMATELY
		77 ;	SOURCE LINE NUMBER 500.
		78 ;	
		79 ;	LINES GENERATED BY MACRO EXPANSION ARE FLAGGED BY A PLUS SIGN ("+")
		80 ;	IMMEDIATELY FOLLOWING THE SOURCE LINE NUMBER.
		81 ;	A NUMBER OF LINES FROM THE VARIOUS MACRO DEFINITIONS WHICH DO NOT
		82 ;	PRODUCE ANY OBJECT CODE ARE PROCESSED BY THE ASSEMBLER
		83 ;	AS THESE MACROS ARE EXPANDED. WHEN THIS IS THE CASE, THESE LINES ARE
		84 ;	SUPPRESSED FROM THE LIST FILE. AS A RESULT, THE LINE NUMBERS ARE
		85 ;	NOT ALWAYS CONSECUTIVE WHERE A MACRO IS BEING INVOKED.
		86 ;	
		87 ;	NOTE:
		88 ;	====
		89 ;	"SOURCE-LINE" REFERS TO THE DECIMAL NUMBERS LEFT OF EACH INSTRUCTION.
		90 ;	AT THE END OF THE LISTING IS AN ASSEMBLY CROSS-REFERENCE TABLE INDICATING
		91 ;	THE SEQUENTIAL SOURCE-LINE NUMBER OF ALL INSTANCES WHERE ANY VARIABLE
		92 ;	IS DEFINED OR REFERENCED. THIS WILL BE OF GREAT ASSISTANCE IN
		93 ;	LOCATING SPECIFIC SUBROUTINES, ETC. IN THE LISTING.
		94 ;	
		95 ;	MNEMONICS COPYRIGHT (C) 1976 INTEL CORPORATION
		96 ;	
		97	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		98 \$	INCLUDE(:F0:ALLOC.MAC)
0000		= 99 ?R1	SET 0
		= 100 ;	
0000		= 101 ?RB0	EQU 0
0001		= 102 ?RB1	EQU 1
0002		= 103 ?RAM	EQU 2
0003		= 104 ?CONST	EQU 3
0004		= 105 ?A	EQU 4 ;ACCUMULATOR VARIABLE TYPE
		= 106 ;	
		= 107 ;	THE FOLLOWING INITIALIZES THE LINKED LIST POINTERS FOR
		= 108 ;	THE REGISTER ALLOCATION AND DEALLOCATION ROUTINES.
		= 109 ;	
0003		= 110 ?B0R2	SET 3
0004		= 111 ?B0R3	SET 4
0005		= 112 ?B0R4	SET 5
0006		= 113 ?B0R5	SET 6
0007		= 114 ?B0R6	SET 7
0008		= 115 ?B0R7	SET 8
		= 116 ;	
0002		= 117 ?B0PNT	SET 2
		= 118 ;	
0003		= 119 ?B1R2	SET 3
0004		= 120 ?B1R3	SET 4
0005		= 121 ?B1R4	SET 5
0006		= 122 ?B1R5	SET 6
0007		= 123 ?B1R6	SET 7
0008		= 124 ?B1R7	SET 8
		= 125 ;	
0002		= 126 ?B1PNT	SET 2
		= 127 ;	
0000		= 128 ORGPG0	SET 000H
0100		= 129 ORGPG1	SET 100H
0200		= 130 ORGPG2	SET 200H
0300		= 131 ORGPG3	SET 300H
0400		= 132 ORGPG4	SET 400H
0500		= 133 ORGPG5	SET 500H
0600		= 134 ORGPG6	SET 600H
0700		= 135 ORGPG7	SET 700H
		= 136 ;	
		= 137 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		= 138 ;	*****
		= 139 ;	
		= 140 ;	START OF ALLOCATION MACROS
		= 141 ;	
		= 142 ;	*****
		= 143 ;	
		= 144 ?RSAVE	MACRO SYMBOL, BANK, PNTVAL
-		= 145 IF	PNTVAL EQ 8
-		= 146	ERROR 2
-		= 147	EXITM
-		= 148	ENDIF
-		= 149 \$	SAVE GEN
-		= 150	SYMBOL SET R&PNTVAL
-		= 151 \$	RESTORE
-		= 152 ?B&BANK&PNT	SET ?B&BANK&R&PNTVAL
		= 153	ENDM
		= 154 ;	
		= 155 ;	
0020		= 156 ?MINDX	SET 20H
		= 157 ;	
		= 158 ?MSAVE	MACRO SYMBOL, LENGTH, ADDR
		= 159 \$	SAVE GEN
-		= 160	SYMBOL EQU ADDR
-		= 161 \$	RESTORE
-		= 162 ?MINDX	SET ?MINDX+LENGTH
		= 163	ENDM
		= 164 ;	
		= 165 MBLOCK	MACRO SYMBOL, LENGTH
-		= 166 ?&SYMBOL	EQU 3
-		= 167	?MSAVE SYMBOL, LENGTH, %?MINDX
		= 168	ENDM
		= 169 ;	
		= 170 DECLARE	MACRO SYMBOL, TYPE
-		= 171 ?&SYMBOL	SET ?&TYPE
-		= 172 IF	?&TYPE EQ 2
-		= 173	?MSAVE SYMBOL, 1, %?MINDX
-		= 174	EXITM
-		= 175	ENDIF
-		= 176 IF	?&TYPE EQ 0
-		= 177	?RSAVE SYMBOL, 0, %?B&PNT
-		= 178	EXITM
-		= 179	ENDIF
-		= 180 IF	?&TYPE EQ 1
-		= 181	?RSAVE SYMBOL, 1, %?B&PNT
-		= 182	EXITM
-		= 183	ENDIF
		= 184	ENDM
		= 185 ;	
		= 186 \$	EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		= 187 ;	
		= 188 ;	REORG MACRO TO RESET THE INSTRUCTION LOCATION COUNTER
		= 189 ;	TO THE FIRST FREE LOCATION ON THE FIRST PAGE MODULE WILL
		= 190 ;	FIT WITHIN.
		= 191	REORG MACRO LOCATION
		= 192	\$SAVE GEN
-		= 193	ORG LOCATION
-		= 194	\$RESTORE
		= 195	ENDM
		= 196 ;	
		= 197 ;	CODEBLK MACRO TO FIND A PAGE OF ROM
		= 198 ;	WHICH THIS BLOCK OF CODE WILL FIT WITHIN
		= 199	CODEBLK MACRO LENGTH
-		= 200	?LENGTH SET LENGTH
-		= 201	IF HIGH(ORGP0+LENGTH-1) EQ 0
-		= 202	REORG %ORGP0
-		= 203	?START SET \$
-		= 204	EXITM
-		= 205	ENDIF
-		= 206	IF HIGH(ORGP1+LENGTH-1) EQ 1
-		= 207	REORG %ORGP1
-		= 208	?START SET \$
-		= 209	EXITM
-		= 210	ENDIF
-		= 211	IF HIGH(ORGP2+LENGTH-1) EQ 2
-		= 212	REORG %ORGP2
-		= 213	?START SET \$
-		= 214	EXITM
-		= 215	ENDIF
-		= 216	IF HIGH(ORGP4+LENGTH-1) EQ 4
-		= 217	REORG %ORGP4
-		= 218	?START SET \$
-		= 219	EXITM
-		= 220	ENDIF
-		= 221	IF HIGH(ORGP5+LENGTH-1) EQ 5
-		= 222	REORG %ORGP5
-		= 223	?START SET \$
-		= 224	EXITM
-		= 225	ENDIF
-		= 226	IF HIGH(ORGP6+LENGTH-1) EQ 6
-		= 227	REORG %ORGP6
-		= 228	?START SET \$
-		= 229	EXITM
-		= 230	ENDIF
-		= 231	IF HIGH(ORGP7+LENGTH-1) EQ 7
-		= 232	REORG %ORGP7
-		= 233	?START SET \$
-		= 234	EXITM
-		= 235	ENDIF
-		= 236	IF HIGH(ORGP3+LENGTH-1) EQ 3
-		= 237	REORG %ORGP3
-		= 238	?START SET \$
-		= 239	EXITM
-		= 240	ENDIF
-		= 241	ERROR 0 ;*** INSUFFICIENT SPACE FOR CODE ON ANY PAGE ***

LOC	OBJ	LINE	SOURCE STATEMENT
		= 242	ENDM
		= 243	;DATABLK INSERTS ONTO PAGE 3
		= 244	DATABLK MACRO LENGTH
-		= 245	?LENGTH SET LENGTH
-		= 246	IF HIGH(ORGP3+LENGTH-1) EQ 3
-		= 247	REORG %ORGP3
-		= 248	?START SET \$
-		= 249	EXITM
-		= 250	ENDIF
-		= 251	ERROR 0 ;*** INSUFFICIENT SPACE FOR DATA BLOCK ON PAGE 3 ***
		= 252	ENDM
		= 253	;SIZE PRINTS A LINE TO THE SOURCE FILE GIVING BLOCK SIZE.
		= 254	; AND UPDATES APPROPRIATE ORGP#
		= 255	?SIZE MACRO BLK,PGE
		= 256	\$SAVE GEN
-		= 257	SIZE SET BLK
-		= 258	;
-		= 259	;*****
-		= 260	IF ?LENGTH LT SIZE
-		= 261	ERROR 0 ;*** SIZE EXCEEDS SPACE CHECKED FOR BY CODEBLK MACRO
-		= 262	ENDIF
-		= 263	IF HIGH(\$-1) NE HIGH(?START)
-		= 264	ERROR 0 ;*** CODE OR DATA BLOCK ROLLED OVER PAGE BOUNDARY ***
-		= 265	ENDIF
-		= 266	\$RESTORE
-		= 267	ORGP&PGE SET \$
		= 268	ENDM
		= 269	;SIZECHK CHECKS SIZE OF PRECEDING BLOCK, PRINTS SIZE TO .LSI FILE.
		= 270	SIZECHK MACRO
-		= 271	?SIZE %(\$-?START),%HIGH(?START)
-		= 272	ENDM
-		= 273	;
-		= 274	;
		= 275	;RSOURCE CODE SPACE ALLOCATION SUMMARY STATEMENT
		= 276	RSOURCE MACRO
		= 277	\$SAVE LIST GEN
-		= 278	PGSIZE SET ORGP0-000H ; BYTES USED ON PAGE 0
-		= 279	PGSIZE SET ORGP1-100H ; BYTES USED ON PAGE 1
-		= 280	PGSIZE SET ORGP2-200H ; BYTES USED ON PAGE 2
-		= 281	PGSIZE SET ORGP3-300H ; BYTES USED ON PAGE 3
-		= 282	PGSIZE SET ORGP4-400H ; BYTES USED ON PAGE 4
-		= 283	PGSIZE SET ORGP5-500H ; BYTES USED ON PAGE 5
-		= 284	PGSIZE SET ORGP6-600H ; BYTES USED ON PAGE 6
-		= 285	PGSIZE SET ORGP7-700H ; BYTES USED ON PAGE 7
-		= 286	\$EJECT
-		= 287	\$RESTORE
-		= 288	ENDM
-		= 289	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		290 ;	
		291 \$	INCLUDE(:F0:MOPCOD.MAC)
		= 292 ;	
		= 293 ; ?FORM1	MACRO FOR GENERALIZING OPCODE INSTRUCTION
		= 294 ;	
		= 295 ?FORM1	MACRO OPCODE, SRC
-		= 296 IF	?&SRC EQ 2
..		= 297 \$	SAVE GEN
-		= 298	MOV R1, #SRC
-		= 299	OPCODE A, @R1
-		= 300 \$	RESTORE
-		= 301	EXITM
-		= 302	ENDIF
..		= 303 IF	?&SRC EQ 0 OR ?&SRC EQ 1
-		= 304 \$	SAVE GEN
..		= 305	OPCODE A, SRC
-		= 306 \$	RESTORE
-		= 307	EXITM
..		= 308	ENDIF
..		= 309 IF	?&SRC EQ 3
..		= 310 \$	SAVE GEN
-		= 311	OPCODE A, #SRC
..		= 312 \$	RESTORE
..		= 313	EXITM
..		= 314	ENDIF
-		= 315	ERROR 1
		= 316	ENDM
		= 317 ;	
		= 318 ; ?FORM2	MACRO FOR GENERALIZING MOVES FROM THE ACC TO A VARIABLE
		= 319 ?FORM2	MACRO DEST
-		= 320 IF	?&DEST EQ 2
-		= 321 \$	SAVE GEN
-		= 322	MOV R1, #DEST
-		= 323	MOV @R1, A
-		= 324 \$	RESTORE
-		= 325	EXITM
-		= 326	ENDIF
..		= 327 IF	?&DEST EQ 0 OR ?&DEST EQ 1
..		= 328 \$	SAVE GEN
-		= 329	MOV DEST, A
-		= 330 \$	RESTORE
-		= 331	EXITM
..		= 332	ENDIF
-		= 333	ERROR 1
		= 334	ENDM
		= 335 ;	
		= 336 ; ?FORM3	MACRO FOR GENERALIZING MOVES FROM THE ACC TO A VARIABLE
		= 337 ;	WHEN IT IS KNOWN THAT R1 (IF NEEDED FOR INDIRECT ADDRESSING)
		= 338 ;	IS ALREADY PRESET.
		= 339 ?FORM3	MACRO DEST
-		= 340 IF	?&DEST EQ 2
-		= 341 \$	SAVE GEN
-		= 342	MOV @R1, A
..		= 343 \$	RESTORE
-		= 344	EXITM

LOC	OBJ	LINE	SOURCE STATEMENT
--		= 345	ENDIF
--		= 346	IF ?&DEST EQ 0 OR ?&DEST EQ 1
--		= 347	\$ SAVE GEN
--		= 348	MOV DEST, A
--		= 349	\$ RESTORE
--		= 350	EXITM
--		= 351	ENDIF
--		= 352	ERROR 1
--		= 353	ENDM
--		= 354	;
--		= 355	;%FORM4 MACRO FOR GENERALIZING 'MOV A, SRC' INSTRUCTION
--		= 356	?FORM4 MACRO SRC
--		= 357	IF ?&SRC EQ 2
--		= 358	\$ SAVE GEN
--		= 359	MOV R1, #SRC
--		= 360	MOV A, @R1
--		= 361	\$ RESTORE
--		= 362	EXITM
--		= 363	ENDIF
--		= 364	IF ?&SRC EQ 0 OR ?&SRC EQ 1
--		= 365	\$ SAVE GEN
--		= 366	MOV A, SRC
--		= 367	\$ RESTORE
--		= 368	EXITM
--		= 369	ENDIF
--		= 370	IF ?&SRC EQ 3
--		= 371	\$ SAVE GEN
--		= 372	MOV A, #SRC
--		= 373	\$ RESTORE
--		= 374	EXITM
--		= 375	ENDIF
--		= 376	ERROR 1
--		= 377	ENDM
--		= 378	;
--		= 379	;%FORM5 MACRO FOR GENERALIZING MOVING A CONSTANT INTO A VARIABLE
--		= 380	?FORM5 MACRO DEST, CONST
--		= 381	IF ?&DEST EQ 0 OR ?&DEST EQ 1 OR ?&DEST EQ 4
--		= 382	\$ SAVE GEN
--		= 383	MOV DEST, #CONST
--		= 384	\$ RESTORE
--		= 385	EXITM
--		= 386	ENDIF
--		= 387	IF ?&DEST EQ 2
--		= 388	\$ SAVE GEN
--		= 389	MOV R1, #DEST
--		= 390	MOV @R1, #CONST
--		= 391	\$ RESTORE
--		= 392	EXITM
--		= 393	ENDIF
--		= 394	ERROR 1
--		= 395	ENDM
--		= 396	;
--		= 397	;%MMOV MACRO GENERALIZED MOVE FROM SRC TO DEST
--		= 398	MMOV MACRO DEST, SRC
--		= 399	IF ?&SRC EQ 3

LOC	OBJ	LINE	SOURCE STATEMENT
-		= 400	?FORM5 DEST, SRC
-		= 401	EXITM
-		= 402	ENDIF
-		= 403	IF ?&DEST EQ 4
-		= 404	?FORM1 MOV, SRC
-		= 405	EXITM
-		= 406	ENDIF
-		= 407	IF ?&SRC EQ 4
-		= 408	?FORM2 DEST
-		= 409	EXITM
-		= 410	ENDIF
-		= 411	?FORM1 MOV, SRC
-		= 412	?FORM2 DEST
-		= 413	ENDM
-		= 414	; ?BINOP MACRO GENERALIZES ARITHMETIC AND LOGICAL OPERATIONS
-		= 415	?BINOP MACRO OPCODE, DEST, SRC
-		= 416	IF ?&DEST EQ 4
-		= 417	?FORM1 OPCODE, SRC
-		= 418	EXITM
-		= 419	ENDIF
-		= 420	IF ?&SRC EQ 4
-		= 421	?FORM1 OPCODE, DEST
-		= 422	?FORM3 DEST
-		= 423	EXITM
-		= 424	ENDIF
-		= 425	?FORM1 MOV, SRC
-		= 426	?FORM1 OPCODE, DEST
-		= 427	?FORM3 DEST
-		= 428	ENDM
-		= 429	; MADD MACRO FOR GENERALIZING ADD INSTRUCTION
-		= 430	MADD MACRO DEST, SRC
-		= 431	?BINOP ADD, DEST, SRC
-		= 432	ENDM
-		= 433	;
-		= 434	; MADDC MACRO FOR GENERALIZING ADDC INSTRUCTION
-		= 435	MADDC MACRO DEST, SRC
-		= 436	?BINOP ADDC, DEST, SRC
-		= 437	ENDM
-		= 438	;
-		= 439	; MANL MACRO FOR GENERALIZING ANL INSTRUCTION
-		= 440	MANL MACRO DEST, SRC
-		= 441	?BINOP ANL, DEST, SRC
-		= 442	ENDM
-		= 443	;
-		= 444	; MORL MACRO FOR GENERALIZING ORL INSTRUCTION
-		= 445	MORL MACRO DEST, SRC
-		= 446	?BINOP ORL, DEST, SRC
-		= 447	ENDM
-		= 448	;
-		= 449	; MXRL MACRO FOR GENERALIZING XRL INSTRUCTION
-		= 450	MXRL MACRO DEST, SRC
-		= 451	?BINOP XRL, DEST, SRC
-		= 452	ENDM
-		= 453	;
-		= 454	; MXCH MACRO FOR GENERALIZING XCH INSTRUCTION

LOC	OBJ	LINE	SOURCE STATEMENT
		= 455	MXCH MACRO DEST, SRC
-		= 456	?BINOP XCH, DEST, SRC
		= 457	ENDM
		= 458 ;	
		= 459	?UNARY MACRO OPCODE, DEST
-		= 460	?FORM1 MOV, DEST
-		= 461	\$SAVE GEN
-		= 462	OPCODE A
-		= 463	\$RESTORE
-		= 464	?FORM3 DEST
		= 465	ENDM
		= 466 ;	
		= 467	MINC MACRO DEST
-		= 468	?UNARY INC, DEST
		= 469	ENDM
		= 470 ;	
		= 471	MDEC MACRO DEST
-		= 472	?UNARY DEC, DEST
		= 473	ENDM
		= 474 ;	
		= 475	MDJNZ MACRO DEST, ADDR
-		= 476	?UNARY DEC, DEST
-		= 477	\$SAVE GEN
-		= 478	JNZ ADDR
-		= 479	\$RESTORE
		= 480	ENDM
		= 481 ;	
		= 482	MRL MACRO DEST
-		= 483	?UNARY RL, DEST
		= 484	ENDM
		= 485 ;	
		= 486	MRR MACRO DEST
-		= 487	?UNARY RR, DEST
		= 488	ENDM
		= 489 ;	
		= 490	MRRC MACRO DEST
-		= 491	?UNARY RRC, DEST
		= 492	ENDM
		= 493 ;	
		= 494	MRLC MACRO DEST
-		= 495	?UNARY RLC, DEST
		= 496	ENDM
		= 497 ;	
		= 498	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		499 ;	
		500 ;=====	
		501 ;=====	
		502 ;	BEGINNING OF PROGRAM PROPER
		503 ;=====	
		504 ;=====	
		505 ;	
		506 ;	
		507 ;*****	
		508 ;	
		509 ;	ALLOCATION OF MP I/O PORTS:
		510 ;	
		511 ;*****	
		512 ;	
		513 ;	BUS ;USED FOR BIDIRECTIONAL ADDRESS AND DATA TRANSFERS
		514 ;	P1 ;USED AS INDIVIDUAL CONTROL OUTPUTS AND BREAK LOGIC
		515 ;	P2 ;HIGH-ORDER ADDRESS AND ADDRESS SPACE SELECTION
		516 ;	
000E		517 PDIGIT EQU	P7 ;USED TO ENABLE CHARACTERS AND STROBE ROWS OF KEYBOARD
000D		518 PSEghi EQU	Y6 ;USED TO TURN ON HI SEGMENTS OF CURRENTLY ENABLED DIGIT
000C		519 PSEGLO EQU	P5 ;PORT FOR LOWER FOUR SEGMENTS
000B		520 PINPUT EQU	P4 ;PORT USED TO SCAN FOR KEY CLOSURES
		521 ;	
		522 ;*****	
		523 ;	
		524 ;	INDIVIDUAL PINS OF PORT 1 USED AS FOLLOWS:
		525 ;	
		526 ;*****	
		527 ;	
0001		528 ENDRAM EQU	00000001B ;P10 - HI ENABLES BREAK ON BREAK RAM OUTPUT SIGNAL
0002		529 ENBLNK EQU	00000010B ;P11 - HI ENABLES BREAK ON RD OR WR TO LINK BY EP
		530	; (NOTE: P11 & P10 BOTH HI ENABLES
		531	; BREAK ON ANY EP INSTRUCTION CYCLE)
0004		532 EPSSTP EQU	00000100B ;P12 - L0 FORCES EP SS INPUT LOW,
		533	; HI GATES BREAKPOINT FLIP-FLOP TO EP SS INPUT.
0008		534 CLRbff EQU	00001000B ;P13 - L0 CLEARS BREAK FLIP-FLOP
		535	; AND ENABLES WR CONTROL TO BREAKPOINT RAM.
0010		536 EPRSET EQU	00010000B ;P14 - HI RESETS EP
0020		537 MODOUT EQU	00100000B ;P15 - L0 WHEN EP IS EXECUTING USER PROGRAM,
		538	; HI WHEN EP FROZEN OR RUNNING OVERLAYS.
0040		539 TTYOUT EQU	01000000B ;P16 - SERIAL OUTPUT TO TTY OR CRT
		540	;P17 - UNUSED
		541 ;	
		542 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		543 ;	*****
		544 ;	
		545 ;	INDIVIDUAL PINS OF PORT 2 USED AS FOLLOWS:
		546 ;	
		547 ;	*****
		548 ;	
		549 ;	P23-P20 ;ADR11-ADR8 FOR ACCESSING PROGRAM OR DATA RAM ARRAY
		550 ;	
0010		551 M0	EQU 00010000B ;P24 -- MEMORY MATRIX CONTROL PIN 0
0020		552 M1	EQU 00100000B ;P25 -- MEMORY MATRIX CONTROL PIN 1
0040		553 MPUSEL	EQU 01000000B ;P26 -- HIGH WHEN MP IN CONTROL OF COMMON MEM ARRAY,
		554	; LOW WHEN EP IN CONTROL.
0080		555 EXPMON	EQU 10000000B ;P27 -- JUMPED TO GROUND FOR STANDARD MONITOR,
		556	; FLOATING WHEN EXPANSION MONITOR PRESENT.
		557 ;	
		558 ;	
		559 ;	WHEN MP IN CONTROL OF MEMORY MATRIX M1-M0 USED AS FOLLOWS:
		560 ;	
		561 ;	M1 M0 MODE
		562 ;	0 0 PROGRAM RAM ARRAY ENABLED FOR READ & WRITE
		563 ;	0 1 DATA RAM ARRAY ENABLED FOR READ & WRITE
		564 ;	1 X LINK REGISTER ENABLED FOR READ, RAM ARRAYS DISABLED.
		565 ;	(NOTE: LINK REGISTER ALWAYS ENABLED FOR MP WRITES)
		566 ;	
		567 ;	WHEN EP IN CONTROL OF MATRIX M1-M0 USED AS FOLLOWS:
		568 ;	
		569 ;	M1 M0 MODE
		570 ;	0 X EP PSEN FETCHES FROM LINK REGISTER (USED TO FORCE OPCODES)
		571 ;	1 0 EP PSEN FETCHES FROM PROGRAM RAM ARRAY,
		572 ;	EP RD & WR CONTROL DATA RAM ARRAY.
		573 ;	1 1 EP PSEN FETCHES FROM PROGRAM RAM ARRAY,
		574 ;	RD & WR CONTROL LINK REGISTER.
		575 ;	
		576	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		577 ;	
		578 ;*****	
		579 ;	
		580 ;	SYSTEM CONSTANT DEFINITIONS:
		581 ;	
		582 ;*****	
		583 ;	
		584 DECLARE CHARNO,CONST	;NUMBER OF DIGITS IN DISPLAY AND ROWS OF KEYS
0008		598 CHARNO EQU	8
		599 ;	
		600 DECLARE NCOLS,CONST	;LESSER DIMENSION OF KEYBOARD MATRIX
0004		614 NCOLS EQU	4
		615 ;	
		616 DECLARE DEBNC,CONST	;NUMBER OF SUCCESSIVE SCANS BEFORE KEY CLOSURE ACCEPTED
0008		630 DEBNC EQU	8
		631 ;	
		632 DECLARE OVSZ,CONST	;SIZE OF LARGEST MINI-MONITOR OVERLAY FOR EP
0017		646 OVSZ EQU	23
		647 ;	
		648 DECLARE BUFLN,CONST	;LENGTH OF HEX FORMAT XMIT BUFFER (MAX RECORD LENGTH)
0010		662 BUFLN EQU	16
		663 ;	
		664 ;*****	
		665 ;	
		666 ;	UTILITY CONSTANT DECLARATIONS
		667 ;	
		668 ;*****	
		669 ;	
		670 DECLARE ZERO,CONST	
0000		684 ZERO EQU	0
		685 DECLARE PLUS1,CONST	
0001		699 PLUS1 EQU	1
		700 DECLARE PLUS3,CONST	
0003		714 PLUS3 EQU	3
		715 DECLARE NEG1,CONST	
FFFF		729 NEG1 EQU	-1
		730 ;	
		731 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		732 ;	
		733 ;*****	
		734 ;	
		735 ;	BANK 0 REGISTER ALLOCATION:
		736 ;	
		737 ;*****	
		738 ;	
		739 DECLARE LDATA,RB0	;DATA USED BY LOGICAL ADDRESSING READ/WRITE UTILITIES
0002		752+ LDATA SET R2	
		756 DECLARE KEY,RB0	;HOLDS KEYCODE RETURNED FROM KBD INPUT ROUTINE.
0003		769+ KEY SET R3	
		773 DECLARE ITMP,RB0	;COUNTER USED AS AN INDEX IN PARSER ROUTINE
0004		786+ ITMP SET R4	
		790 DECLARE CHKSUM,RB0	;CHECKSUM OF DATA BYTES TRANSMITTED IN HEX FILE FORMAT
0005		803+ CHKSUM SET R5	
		807 DECLARE DSPTMP,RB0	;TEMPORARY STORAGE FOR DISPLAY PATTERNS IN 'DSPACC'
0006		820+ DSPTMP SET R6	
		824 DECLARE XPCODE,RB0	;EXPANSION MONITOR ROUTINE CODE NUMBER
0007		837+ XPCODE SET R7	
		841 ;	
		842 ;*****	
		843 ;	
		844 ;	BANK 1 REGISTER ALLOCATION
		845 ;	
		846 ;*****	
		847 ;	
		848 DECLARE ROTPAT,RB1	;USED TO HOLD INPUT PATTERN BEING ROTATED THROUGH CY
0002		865+ ROTPAT SET R2	
		869 DECLARE ROTCNT,RB1	;COUNTS NUMBER OF BITS ROTATED THROUGH CY
0003		886+ ROTCNT SET R3	
		890 DECLARE LASTKY,RB1	;HOLDS KEY POSITION OF LAST KEY DEPRESSION DETECTED
0004		907+ LASTKY SET R4	
		911 DECLARE CURDIG,RB1	;HOLDS POSITION OF NEXT CHARACTER TO BE DISPLAYED
0005		928+ CURDIG SET R5	
		932 DECLARE KEYFLG,RB1	;FLAG TO DETECT WHEN ALL KEYS ARE RELEASED
0006		949+ KEYFLG SET R6	
		953	; (REGISTER 7 NOT USED FOR PRIMARY MONITOR)
		954 ;	
		955 ;*****	
		956 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		957 ;	
		958 ; *****	
		959 ;	
		960 ; DATA RAM ALLOCATION	
		961 ;	
		962 ; *****	
		963 ;	
		964 DECLARE EPACC, RAM ; STORAGE IN MP FOR EP ACCUMULATOR	
0020		969+ EPACC EQU 32	
		973 DECLARE EPPSW, RAM ; STORAGE IN MP FOR EP PROGRAM STATUS WORD	
0021		978+ EPPSW EQU 33	
		982 DECLARE EPTIMR, RAM ; STORAGE IN MP FOR EP TIMER/COUNTER REGISTER	
0022		987+ EPTIMR EQU 34	
		991 DECLARE EPR0, RAM ; STORAGE IN MP FOR EP REGISTER 0 OF BANK 0	
0023		996+ EPR0 EQU 35	
		1000 DECLARE EPPCLO, RAM ; STORAGE IN MP FOR LOW BYTE OF EP PROGRAM COUNTER	
0024		1005+ EPPCLO EQU 36	
		1009 DECLARE EPPCHI, RAM ; STORAGE IN MP FOR HIGH NIBBLE OF EP PROGRAM COUNTER	
0025		1014+ EPPCHI EQU 37	
		1018 DECLARE HBITLO, RAM ; PARAMETER 1 FOR SERIAL LINK DATA RATE GENERATOR	
0026		1023+ HBITLO EQU 38	
		1027 DECLARE HBITHI, RAM ; PARAMETER 2 FOR SERIAL LINK DATA RATE GENERATOR	
0027		1032+ HBITHI EQU 39	
		1036 DECLARE DSPTIM, RAM ; PARAMETER FOR AUTO-STEP AND AUTO-BREAK SEQUENCING RATE	
0028		1041+ DSPTIM EQU 40	
		1045 DECLARE VERSNO, RAM ; MONITOR VERSION NUMBER	
0029		1050+ VERSNO EQU 41	
		1054 DECLARE HREGA, RAM ; (UNUSED)	
002A		1059+ HREGA EQU 42	
		1063 DECLARE HREGD, RAM ; (UNUSED)	
002B		1068+ HREGD EQU 43	
		1072 DECLARE HREGC, RAM ; (UNUSED)	
002C		1077+ HREGC EQU 44	
		1081 DECLARE HREGI, RAM ; (UNUSED)	
002D		1086+ HREGI EQU 45	
		1090 DECLARE HREGF, RAM ; (UNUSED)	
002E		1095+ HREGF EQU 46	
		1099 DECLARE HREGG, RAM ; (UNUSED)	
002F		1104+ HREGG EQU 47	
		1108 DECLARE SMAILO, RAM ; PRIMARY COMMAND STARTING MEMORY ADDRESS (LOW BYTE)	
0030		1113+ SMAILO EQU 48	
		1117 DECLARE SMAHI, RAM ; PRIMARY COMMAND STARTING MEMORY ADDRESS (HIGH BYTE)	
0031		1122+ SMAHI EQU 49	
		1126 DECLARE EMAILO, RAM ; PRIMARY COMMAND ENDING MEMORY ADDRESS (LOW BYTE)	
0032		1131+ EMAILO EQU 50	
		1135 DECLARE EMAHI, RAM ; PRIMARY COMMAND ENDING MEMORY ADDRESS (HIGH BYTE)	
0033		1140+ EMAHI EQU 51	
		1144 DECLARE MEMILO, RAM ; THIRD PARSER PARAMETER & HEX RECORD ADDRESS (LOW)	
0034		1149+ MEMILO EQU 52	
		1153 DECLARE MEMHI, RAM ; THIRD PARSER PARAMETER & HEX RECORD ADDRESS (HIGH)	
0035		1158+ MEMHI EQU 53	
		1162 DECLARE BCODE, RAM ; PRIMARY COMMAND NUMBER FROM PARSER TABLES (0-6)	
0036		1167+ BCODE EQU 54	
		1171 DECLARE TYPE, RAM ; PRIMARY COMMAND MODIFIER/OPTION (0-5)	
0037		1176+ TYPE EQU 55	

LOC	OBJ	LINE	SOURCE STATEMENT
		1180	DECLARE NUMCON, RAM ; MAX. NUMBER OF PARAMETERS ALLOWED FOR SELECTED COMMAND
0038		1185+	NUMCON EQU 56
		1189	DECLARE OPTION, RAM ; INDEX POINTER USED IN SEARCHING PARSER TABLES
0039		1194+	OPTION EQU 57
		1198	DECLARE NEXTPL, RAM ; CHARACTER POSITION FOR DISPLAY UTILITIES TO WRITE NEXT
003A		1203+	NEXTPL EQU 58
		1207	DECLARE KBDDBUF, RAM ; POSITION OF KEY DEBOUNCED BY SCANNING SUBROUTINE
003B		1212+	KBDDBUF EQU 59
		1216	DECLARE KEYLOC, RAM ; INCREMENTED AS SUCCESSIVE KEY LOCATIONS SCANNED
003C		1221+	KEYLOC EQU 60
		1225	DECLARE NREPTS, RAM ; KEEPS TRACK OF SUCCESSIVE READS OF SAME KEYSTROKE
003D		1230+	NREPTS EQU 61
		1234	DECLARE ASAVE, RAM ; HOLDS ACCUMULATOR VALUE DURING SERVICE ROUTINE
003E		1239+	ASAVE EQU 62
		1243	DECLARE RDELAY, RAM ; COUNTER DECREMENTED WHEN AUTO-STEP DELAY IN PROGRESS
003F		1248+	RDELAY EQU 63
		1252	DECLARE STRTMP, RAM ; INDEX POINTER FOR DISPLAY CHARACTER STRING ACCESSING
0040		1257+	STRTMP EQU 64
		1261	DECLARE BUFCNT, RAM ; COUNT OF DATA BYTES IN HEX FORMAT RECORD BUFFER
0041		1266+	BUFCNT EQU 65
		1270	DECLARE RECTYP, RAM ; TYPE OF HEX FORMAT RECORD (0 OR 1)
0042		1275+	RECTYP EQU 66
		1279	DECLARE B, RAM ; BIT COUNTER FOR ASCII SERIAL I/O UTILITY SUBROUTINES
0043		1284+	B EQU 67
		1288	DECLARE REGC, RAM ; CHARACTER BEING SHIFTED DURING SERIAL I/O PROCESS
0044		1293+	REGC EQU 68
		1297	DECLARE H, RAM ; COUNTER IN SOFTWARE DELAY DATA RATE GENERATOR
0045		1302+	H EQU 69
		1306	;
		1307	MBLOCK SEGMAP, CHARNO ; REGISTER ARRAY FOR DISPLAY PATTERNS
0046		1311+	SEGMAP EQU 70
		1314	;
		1315	MBLOCK OVBUFF, OVSIZE ; LOW-ORDER USER PROGRAM DURING MINI-MONITOR OVERLAYS
004E		1319+	OVBUFF EQU 78
		1322	;
		1323	MBLOCK HEXBUF, BUFLen ; ALLOCATE BLOCK OF RAM FOR USE AS HEX RECORD BUFFER
0065		1327+	HEXBUF EQU 101
		1330	;
		1331	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		1332	DATABLK 40
0300		1337+	ORG 768
		1341	; INVALS TABLE OF CONSTANTS TO BE LOADED INTO MP INTERNAL RAM VARIABLES
		1342	; AS PART OF SYSTEM INITIALIZATION PROCEDURE:
		1343	;
		1344	;
		1345	;
			=====
0300 00		1346	INVALS: DB 00H ; ROTPAT RB1
0301 00		1347	DB 00H ; ROTCNT RB1
0302 00		1348	DB 00H ; LASTKY RB1
0303 00		1349	DB CHARNO ; CURDIG RB1
0304 00		1350	DB 00H ; KEYFLG RB1
0305 00		1351	DB 00H ; <REG7> RB1
0306 00		1352	DB 00H ; EPACC RAM
0307 01		1353	DB 01H ; EPPSW RAM
0308 00		1354	DB 00H ; EPTIMR RAM
0309 00		1355	DB 00H ; EPR0 RAM
030A 00		1356	DB 00H ; EPPCLO RAM
030B 00		1357	DB 00H ; EPPCHI RAM
030C 93		1358	DB 93H ; HBITLO RAM
030D 04		1359	DB 04H ; HBITHI RAM
030E 20		1360	DB 20H ; DSPTIM RAM
030F 25		1361	DB 25H ; VERSNO RAM
0310 00		1362	DB 00H ; HREGA RAM
0311 00		1363	DB 00H ; HREGB RAM
0312 00		1364	DB 00H ; HREGC RAM
0313 00		1365	DB 00H ; HREGD RAM
0314 00		1366	DB 00H ; HREG E RAM
0315 00		1367	DB 00H ; HREGF RAM
0316 00		1368	DB 00H ; SMAILO RAM
0317 00		1369	DB 00H ; SMAHI RAM
0318 FF		1370	DB 0FFH ; EMAILO RAM
0319 0F		1371	DB 0FH ; EMAHI RAM
031A 00		1372	DB 00H ; MEMILO RAM
031B 00		1373	DB 00H ; MEMHI RAM
031C 00		1374	DB 00H ; BCODE RAM
031D 04		1375	DB 04H ; TYPE RAM
031E 01		1376	DB 01H ; NUMCON RAM
031F 00		1377	DB 00H ; OPTION RAM
0320 00		1378	DB CHARNO ; NEXTPL RAM
0321 FF		1379	DB 0FFH ; KBDUF RAM
0322 00		1380	DB 00H ; KEYLOC RAM
0023		1381	NOVALS EQU \$- INVALS
		1382	SIZECHK
0023		1385+	SIZE SET 35
		1386+	;
		1387+	*****
		1396	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		1397 \$	INCLUDE(:F0:PARSER.MOD)
		=1398	CODEBLK 45
0000		=1403+	ORG 0
		=1407 ;	INIT INITIALIZES PROCESSOR REGISTERS
		=1408 ;	AND RAM LOCATIONS TO DEFINED VALUES.
0000 C5		=1409 INIT:	SEL R00
0001 BF00		=1410	MOV XPCODE, #0
0003 74D1		=1411	CALL XPTST
0005 27		=1412	CLR A
0006 3D		=1413	MOVD PSEGLO, A
0007 3E		=1414	MOVD PSEGH1, A
0008 B81A		=1415	MOV R0, #1AH ; START AT RD1 <REG2> = RAM LOC 1AH
000A B923		=1416	MOV R1, #LOW NOVALS
000C BA00		=1417	MOV R2, #LOW INVALS
000E FA		=1418 INITLP:	MOV A, R2
000F E3		=1419	MOVP3 A, @A
0010 A0		=1420	MOV @R0, A
0011 18		=1421	INC R0
0012 1A		=1422	INC K2
0013 E90E		=1423	DJNZ K1, INITLP
0015 55		=1424	STRT T
0016 744F		=1425	CALL EPERK
0018 68BB		=1426	MOV R0, #LOW(OV1BAS+OVSIZE)
001A 74CA		=1427	CALL OVLOAD
001C 54E5		=1428	CALL COMFIL
001E B937		=1429	MOV R1, #TYPE
0020 11		=1430	INC @R1
0021 34F2		=1431	CALL INCSMA
0023 54E5		=1432	CALL COMFIL
0025 99EF		=1433	ANL P1, #(NOT EPRSET) ; REMOVE EP RESET SIGNAL.
0027 0429		=1434	JMP MAIN
		=1435 ;	
		=1436	SIZECHK
0029		=1439+	SIZE SET 41
		=1440+;	
		=1441+;	*****
		=1450	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=1451 ;	
		=1452 ;	KEYBOARD LAYOUT:
		=1453 ;	=====
		=1454 ;	
		=1455 ;	
		=1456 ;	LIST !!GO/RESET!! GO !!EXAM/CHA!! C !! D !! E !! F !!
		=1457 ;	
		=1458 ;	
		=1459 ;	
		=1460 ;	
		=1461 ;	!!PROG BRK!!PROG MEM!!REGISTER!!
		=1462 ;	UPLOAD !! 8 !! 9 !! A !! B !!
		=1463 ;	!!AUTO STP!!SING STP!! NO BRK !!
		=1464 ;	
		=1465 ;	
		=1466 ;	!!DATA BRK!!DATA MEM!!
		=1467 ;	DNLOAD !! 4 !! 5 !! 6 !! 7 !!
		=1468 ;	!!AUTO BRK!!WITH BRK!!
		=1469 ;	
		=1470 ;	
		=1471 ;	FILL !!HARD REG!! NEXT/, !! END/ !! 0 !! 1 !! 2 !! 3 !!
		=1472 ;	
		=1473 ;	
		=1474 ;	
		=1475 ;	
		=1476 ;	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=1477 ;	
		=1478 ;	THE FOLLOWING EQUATES DETERMINES HOW THE PARSER INTERPRETS
		=1479 ;	VALUES RETURNED BY THE KEYBOARD SCANNING INPUT ROUTINE
		=1480 ;	WHEN THE VARIOUS KEYS OF THE KEYBOARD ARE PRESSED.
		=1481 ;	
		=1482 ;	
		=1483 ; KEY0 EQU 00H	VALUE RETURNED FOR EACH KEY OF KEYBOARD MATRIX
		=1484 ; KEY1 EQU 01H	BY KEYBOARD SCANNING SUBROUTINE "KDDIN".
		=1485 ; KEY2 EQU 02H	
		=1486 ; KEY3 EQU 03H	+---+---+---+---+ +---+---+---+---+
		=1487 ; KEY4 EQU 04H	! 1C ! 1D ! 1E ! 1F ! ! 0C ! 0D ! 0E ! 0F !
		=1488 ; KEY5 EQU 05H	+---+---+---+---+ +---+---+---+---+
		=1489 ; KEY6 EQU 06H	! 18 ! 19 ! 1A ! 1B ! ! 08 ! 09 ! 0A ! 0B !
		=1490 ; KEY7 EQU 07H	+---+---+---+---+ +---+---+---+---+
		=1491 ; KEY8 EQU 08H	! 14 ! 15 ! 16 ! 17 ! ! 04 ! 05 ! 06 ! 07 !
		=1492 ; KEY9 EQU 09H	+---+---+---+---+ +---+---+---+---+
		=1493 ; KEYA EQU 0AH	! 10 ! 11 ! 12 ! 13 ! ! 00 ! 01 ! 02 ! 03 !
		=1494 ; KEYB EQU 0BH	+---+---+---+---+ +---+---+---+---+
		=1495 ; KEYC EQU 0CH	
		=1496 ; KEYD EQU 0DH	
		=1497 ; KEYE EQU 0EH	
		=1498 ; KEYF EQU 0FH	
0010		=1499 KEYFIL EQU 10H	:[FILL COMMAND]
0012		=1500 KEYNXT EQU 12H	:[NEXT/,]
0013		=1501 KEYEND EQU 13H	:[END/,]
0014		=1502 KEYREL EQU 14H	:[DOWNLOAD COMMAND]
0015		=1503 KEYPAT EQU 15H	:[AUTOBREAK MODIFIER]
0016		=1504 KEYDM EQU 16H	:[DATA MEMORY MODIFIER]
0017		=1505 KEYCLR EQU 17H	:[CLEAR/PREVIOUS]
0018		=1506 KEYREC EQU 18H	:[UPLOAD COMMAND]
0019		=1507 KEYTRA EQU 19H	:[AUTOSTEP MODIFIER]
001A		=1508 KEYPM EQU 1AH	:[PROGRAM MEMORY MODIFIER]
001B		=1509 KEYREG EQU 1BH	:[REGISTER MEMORY MODIFIER]
001C		=1510 KEYLST EQU 1CH	:[FORMATTED DATA OUTPUT COMMAND]
001D		=1511 KGORES EQU 1DH	:[GO FROM RESET STATE COMMAND]
001E		=1512 KEYGO EQU 1EH	:[GO COMMAND]
001F		=1513 KEYMOD EQU 1FH	:[EXAMINE/MODIFY COMMAND]
0008		=1514 KSETB EQU 0BH	:[SET BREAKPOINT COMMAND]
000C		=1515 KCLRB EQU 0CH	:[CLEAR BREAKPOINT COMMAND]
		=1516 ;	
		=1517 ;	
0019		=1518 PBK EQU 19H	:[PROGRAM BREAKPOINT MEMORY MODIFIER]
0015		=1519 DBK EQU 15H	:[DATA BREAKPOINT MEMORY MODIFIER]
0011		=1520 RINT EQU 11H	:[HARDWARE REGISTER MEMORY MODIFIER]
001B		=1521 NOBRK EQU 1BH	:[WITHOUT BREAKPOINTS MODIFIER]
0016		=1522 WBRK EQU 16H	:[WITH BREAKPOINTS ENABLED MODIFIER]
001A		=1523 SING EQU 1AH	:[SINGLE STEP MODIFIER]
		=1524 ;	
		=1525 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		=1526	CODEBLK 160
0029		=1531+	ORG 41
		=1535 ;	MAIN OUTPUT_MESSAGE(COMMAND_PROMPT)
		=1536 ;	CALL INPUT_BYTE(KEY)
		=1537 ;	MAIN2 IF THE KEY=END GO TO MAIN
		=1538 ;	
0029	B001	=1539	MAIN: MOV XPCODE, #1
002B	74D1	=1540	CALL XPTEST
002D	2301	=1541	MOV R, #1
002F	3400	=1542	CALL OUTUTL
0031	14EC	=1543	CALL INPKEY
0033	FB	=1544	MAIN2: MOV R, KEY
0034	D313	=1545	XRL R, #KEYEND
0036	C629	=1546	JZ MAIN
		=1547 ;	
		=1548 ;	FINDOP FIND OUT IF THE KEY PRESSED IS A LEGITIMATE COMMAND INITIATOR:
		=1549 ;	ITMP:=CTAB
		=1550 ;	BCODE:=TYPE:=0
		=1551 ;	WHILE CTAB(ITMP)<>0 /CTAB EXHAUSTED/
		=1552 ;	IF CTAB(ITMP)=KEY GOTO MAINA /COMMAND ENTRY FOUND IN CTAB/
		=1553 ;	ELSE ITMP:=ITMP+COMMAND_ENTRY_SIZE
		=1554 ;	BCODE:=BCODE+1
		=1555 ;	ENDWHILE
		=1556 ;	GOTO ERROR
0038	BC23	=1557	MOV ITMP, #CTAB
		=1558	MMOV BCODE, ZERO
003A	B936	=1569+	MOV R1, #BCODE
003C	B100	=1570+	MOV @R1, #ZERO
		=1574	MMOV TYPE, ZERO
003E	B937	=1585+	MOV R1, #TYPE
0040	B100	=1586+	MOV @R1, #ZERO
0042	FC	=1590	FINDOP: MOV R, ITMP
0043	E3	=1591	MOV#3 R, @R
0044	B2BC	=1592	JBS MERROR
0046	D0	=1593	XRL R, KEY
0047	C652	=1594	JZ MAINA
0049	FC	=1595	MOV R, ITMP
004A	0303	=1596	ADD R, #COMSIZ
004C	AC	=1597	MOV ITMP, R
004D	B936	=1598	MOV R1, #BCODE
004F	11	=1599	INC @R1
0050	0442	=1600	JMP FINDOP
		=1601 ;	
		=1602 ;	OUTPUT_MESSAGE(STRCOM(BCODE)) /*PROMPT FOR THE CURRENT COMMAND*/
		=1603 ;	I:=I+1
		=1604 ;	OPTION:=MEM(I)
		=1605 ;	I:=I+1
		=1606 ;	NO_OF_PARAMETERS:=MEM(I)
		=1607 ;	I:=3
		=1608 ;	
		=1609	MAINA: MMOV R, BCODE
0052	B936	=1618+	MOV R1, #BCODE
0054	F1	=1619+	MOV R, @R1
0055	031D	=1623	ADD R, #STRCOM
0057	3402	=1624	CALL OUTCLR

LOC	OBJ	LINE	SOURCE STATEMENT
0059	1C	=1625	INC ITMP
005A	FC	=1626	MOV A, ITMP
005B	E3	=1627	MOVFP3 A, @A ; GET OPTION POINTER
		=1628	MMOV OPTION, A
005C	B939	=1641+	MOV R1, #OPTION
005E	A1	=1642+	MOV @R1, A
005F	1C	=1646	INC ITMP
0060	FC	=1647	MOV A, ITMP
0061	E3	=1648	MOVFP3 A, @A ; GET NO OF PARAMETERS
		=1649	MMOV NUMCON, A
0062	B938	=1662+	MOV R1, #NUMCON
0064	A1	=1663+	MOV @R1, A
		=1667 ;	
		=1668 ;	PARAMETER_BUFFER(0=>5):=0
		=1669 ;	
0065	B906	=1670	MOV R1, #6 ; EACH PARAM IS 2 BYTES
0067	B030	=1671	MOV R0, #SMALO ; START OF PARAM BUFFERS
0069	B000	=1672 MAINB:	MOV @R0, #00H
006B	18	=1673	INC R0
006C	E969	=1674	DJNZ R1, MAINB
006E	14EC	=1675	CALL INPKEY
		=1676 ;	
		=1677 ;	WHILE KEY<>MEM(OPTION+TYPE)[6-0] DO
		=1678 ;	IF MEM(OPTION+TYPE)[7]=1 GOTO MAIND1
		=1679 ;	TYPE:=TYPE+1
		=1680 ;	ENDWHILE
		=1681 ;	
		=1682	MMOV ITMP, OPTION
0070	B939	=1690+	MOV R1, #OPTION
0072	F1	=1699+	MOV A, @R1
0073	0C	=1712+	MOV ITMP, A
0074	1C	=1715	INC ITMP
		=1716 MAINC1:	MMOV A, ITMP
0075	FC	=1732+	MOV A, ITMP
0076	E3	=1736	MOVFP3 A, @A
0077	97	=1737	CLR C
0078	F7	=1738	RLC A
0079	77	=1739	RR A ; STRIP BIT SEVEN INTO CARRY
007A	DB	=1740	XRL A, KEY
007B	C693	=1741	JZ MAIND
007D	F687	=1742	JC MAIND1
		=1743	MINC TYPE
007F	B937	=1748+	MOV R1, #TYPE
0081	F1	=1749+	MOV A, @R1
0082	17	=1753+	INC A
0083	A1	=1758+	MOV @R1, A
0084	1C	=1761	INC ITMP
0085	0475	=1762	JMP MAINC1
		=1763 ;	
		=1764 ;	MODIFIER NOT FOUND SO RESET TYPE INDEX TO DEFAULT CASE (ZERO).
		=1765 ;	
		=1766 MAIND1:	MMOV TYPE, ZERO
0087	B937	=1777+	MOV R1, #TYPE
0089	B100	=1778+	MOV @R1, #ZERO
		=1782	MMOV A, OPTION

LOC	OBJ	LINE	SOURCE STATEMENT
0088	B939	=1791+	MOV R1, #OPTION
008D	F1	=1792+	MOV A, @R1
008E	E3	=1796	MOVFP3 A, @A
008F	3404	=1797	CALL OUTMSG
0091	049E	=1798	JMP MAINB0
		=1799 ;	
		=1800 ;	CALL OUTPUT_MESSAGE(MODIFIER)
		=1801 MAIND:	MMOV A, OPTION
0093	B939	=1810+	MOV R1, #OPTION
0095	F1	=1811+	MOV A, @R1
0096	E3	=1815	MOVFP3 A, @A
		=1816	MADD A, TYPE
0097	B937	=1822+	MOV R1, #TYPE
0099	61	=1823+	ADD A, @R1
009A	3404	=1827	CALL OUTMSG
009C	14EC	=1828	CALL INPKEY
		=1829 ;	
009E	DC00	=1830 MAINB0:	MOV ITEMP, #0
00A0	2330	=1831 MAINB1:	MOV A, #SMILO
00A2	6C	=1832	ADD A, ITEMP
00A3	6C	=1833	ADD A, ITEMP
00A4	A8	=1834	MOV R0, A
00A5	14C0	=1835	CALL INPADR
00A7	F6BA	=1836	JC CMDINT
00A9	1C	=1837	INC ITEMP
00AA	B938	=1838	MOV R1, #NUMCON
00AC	F1	=1839	MOV A, @R1
00AD	07	=1840	DEC A
00AE	A1	=1841	MOV @R1, A
00AF	C6BA	=1842	JZ CMDINT
00B1	FB	=1843	MOV A, KEY
00B2	D313	=1844	XRL A, #KEYEND
00B4	C6BA	=1845	JZ CMDINT
00B6	14EC	=1846	CALL INPKEY
00B8	04A0	=1847	JMP MAINB1
		=1848 ;	
		=1849 ;	CMDINT ENTER THE COMMAND PROCESSOR WITH:
		=1850 ;	BASE_CODE=THE MAIN COMMAND TYPE
		=1851 ;	TYPE=SUBCOMMAND TYPE
		=1852 ;	PARAMETER(1)=FIRST ADDRESS
		=1853 ;	PARAMETER(2)=SECOND ADDRESS
		=1854 ;	PARAMETER(3)=DATA
00DA	4400	=1855 CMDINT:	JMP IMPLM
		=1856 ;	
		=1857 ;	MERROR ERROR ENCOUNTERED IN MAIN PARSING ROUTINE.
00BC	BA01	=1858 MERROR:	MOV LDATA, #1
00BE	249A	=1859	JMP PERROR
		=1860	SIZECHK
0097		=1863+	SIZE SET 151
		=1864+;	
		=1865+;	*****
		=1874	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=1875	DATABLK 50
0323		=1880+	ORG 803
		=1884 ;	
		=1885 ;	*****
		=1886 ;	
		=1887 ;	TABLES FOR PARSER
		=1888 ;	
		=1889 ;	*****
		=1890 ;	
		=1891 ;	THE CTAB TABLE CONTAINS <COMSIZ> ENTRIES FOR EACH COMMAND. THE MEANING
		=1892 ;	OF THE ENTRIES IS AS FOLLOWS:
		=1893 ;	
		=1894 ;	ENTRY 0. COMMAND KEY TO INITIATE
		=1895 ;	ENTRY 1. POINTER TO THE LIST OF OPTIONS APPLICABLE TO THIS COMMAND
		=1896 ;	ENTRY 2. NUMBER OF NUMERIC PARAMETERS REQUIRED BY THE COMMAND
		=1897 ;	
0023		=1898	CTAB EQU \$ AND OFFH
0003		=1899	COMSIZ EQU 3
		=1900 ;	
0323 1F		=1901	DB KEYMOD, LOW OPTAB1, 1 ; EXAM
0324 3F		=	
0325 01		=	
0326 1E		=1902	DB KEYGO, LOW OPTAB3, 1 ; GO
0327 49		=	
0328 01		=	
0329 10		=1903	DB KEYFIL, LOW OPTAB1, 3 ; FILL
032A 3F		=	
032B 03		=	
032C 1C		=1904	DB KEYLST, LOW OPTAB1, 2 ; DUMP
032D 3F		=	
032E 02		=	
032F 18		=1905	DB KEYREC, LOW OPTAB1, 2 ; RECORD
0330 3F		=	
0331 02		=	
0332 14		=1906	DB KEYREL, LOW OPTAB1, 0 ; RELOAD
0333 3F		=	
0334 00		=	
0335 00		=1907	DB KSETB, LOW OPTAB2, 1 ; SETBRK
0336 46		=	
0337 01		=	
0338 0C		=1908	DB KCLRB, LOW OPTAB2, 1 ; CLRBRK
0339 46		=	
033A 01		=	
033B 1D		=1909	DB KGORES, LOW OPTAB3, 0 ; GO FROM RESET STATE
033C 49		=	
033D 00		=	
033E FF		=1910	DB OFFH ; ESCOP
		=1911 ;	
		=1912	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=1913 ;	
		=1914 ;	THE OPTION TABLE GIVES THE VARIOUS OPTIONS ALLOWED FOR EACH
		=1915 ;	BASIC COMMAND, AS FOLLOWS:
		=1916 ;	
		=1917 ;	ENTRY 0. START OF TABLE OF MODIFIER RESPONSES.
		=1918 ;	ENTRY 1+. ALLOWED MODIFIER KEYSTROKES CORRESPONDING TO OPTIONS 0-5.
		=1919 ;	NOTE THAT THE LAST BYTE IN EACH OPTION GROUP HAS BIT
		=1920 ;	SEVEN SET TO INDICATE THE END.
		=1921 ;	
033F	26	=1922 OPTAB1: DB	STRMEM
0340	1A	=1923 DB	KEYPM, KEYDM, KEYREG, RINT
0341	16	=	
0342	1B	=	
0343	11	=	
0344	19	=1924 DB	PBRK, DBRK OR 00H
0345	95	=	
0346	26	=1925 OPTAB2: DB	STRMEM
0347	1A	=1926 DB	KEYPM, KEYDM OR 00H
0348	96	=	
0349	2C	=1927 OPTAB3: DB	STRGOC
034A	1B	=1928 DB	NOBRK, WBRK, SING
034B	16	=	
034C	1A	=	
034D	15	=1929 DB	KEYPAT, KEYTRA OR 00H
034E	99	=	
		=1930	SIZECHK
002C		=1933+ SIZE SET 44	
		=1934+;	
		=1935+; ****	
		=1944 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		=1945	CODEBLK 130
0100		=1955+	ORG 256
		=1959 ;	OUTUTL OUTPUT ONE OF FOUR UTILITY DISPLAY PROMPTS (LEFT JUSTIFIED)
		=1960 ;	ACCORDING TO ACC CONTENTS (0-3).
		=1961 ;	OUTCLR CLEAR DISPLAY AND OUTPUT CHARACTER STRING STARTING
		=1962 ;	AT THE ADDRESS POINTED TO BY BYTE AT ADDRESS IN ACCUMULATOR.
		=1963 ;	OUTMSG SUBROUTINE TO COPY A STRING OF BIT PATTERNS FROM ROM TO THE
		=1964 ;	DISPLAY REGISTERS.
		=1965 ;	STRING SELECTED IS DETERMINED BY ACC WHEN CALLED.
		=1966 ;	ON ENTERING OUTMSG, ACC CONTENTS ARE USED TO ADDRESS A BYTE IN A
		=1967 ;	LOOKUP TABLE ON THE CURRENT PAGE WHICH CONTAINS THE ADDRESS OF
		=1968 ;	A STRING OF SEGMENT PATTERN DATA BYTES TO BE PRINTED ONTO THE
		=1969 ;	DISPLAY.
		=1970 ;	THE END OF THE STRING IS INDICATED WHEN BIT7 =1
		=1971 ;	CALLS SUBROUTINE 'WDISP'
		=1972 ;	TO ACTUALLY EFFECT WRITING INTO THE DISPLAY REGISTERS.
0100 0319		=1973	OUTUTL: ADD A, #STRUTL
0102 B4F1		=1974	OUTCLR: CALL CLEAR
0104 A3		=1975	OUTMSG: MOVP A, @A
		=1976	MMOV STRTMP, A
0105 E940		=1989+	MOV R1, #STRTMP
0107 A1		=1990+	MOV @R1, A
		=1994	PRNT2: MMOV A, STRTMP ; LOAD NEXT CHARACTER LOCATION
0108 E940		=2003+	MOV R1, #STRTMP
010A F1		=2004+	MOV A, @R1
010B A3		=2008	MOVP A, @A ; LOAD BIT PATTERN INDIRECT
010C F217		=2009	JB7 PRNT1
010E D4D8		=2010	CALL WDISP ; OUTPUT TO NEXT CHARACTER POSITION
		=2011	MINC STRTMP ; INDEX POINTER
0110 D940		=2016+	MOV R1, #STRTMP
0112 F1		=2017+	MOV A, @R1
0113 17		=2021+	INC A
0114 A1		=2026+	MOV @R1, A
0115 2408		=2029	JMP PRNT2
0117 C4D8		=2030	PRNT1: JMP WDISP ; DONE
		=2031 ;	
0019		=2032	STRUTL EQU LOW \$
0119 31		=2033	DB LOW(DERR0R) ; UTILITY MESSAGE 0 ADDRESS
011A 37		=2034	DB LOW(DSGNON) ; UTILITY MESSAGE 1 ADDRESS
011B 3E		=2035	DB LOW(DRUN) ; UTILITY MESSAGE 2 ADDRESS
011C 44		=2036	DB LOW(DBPNT) ; UTILITY MESSAGE 3 ADDRESS
001D		=2037	STRCOM EQU LOW \$
011D 46		=2038	DB LOW(DMOD) ; BASIC COMMAND 0 RESPONSE ADDRESS
011E 49		=2039	DB LOW(DGO) ; BASIC COMMAND 1 RESPONSE ADDRESS
011F 4B		=2040	DB LOW(DFILL) ; BASIC COMMAND 2 RESPONSE ADDRESS
0120 4E		=2041	DB LOW(DLST) ; BASIC COMMAND 3 RESPONSE ADDRESS
0121 51		=2042	DB LOW(DREC) ; BASIC COMMAND 4 RESPONSE ADDRESS
0122 54		=2043	DB LOW(DREL) ; BASIC COMMAND 5 RESPONSE ADDRESS
0123 57		=2044	DB LOW(DSB) ; BASIC COMMAND 6 RESPONSE ADDRESS
0124 5A		=2045	DB LOW(DCB) ; BASIC COMMAND 7 RESPONSE ADDRESS
0125 5D		=2046	DB LOW(DGR) ; BASIC COMMAND 8 RESPONSE ADDRESS
0026		=2047	STRMEM EQU LOW \$
0126 5F		=2048	DB LOW(DPRMEM) ; DATA TYPE MODIFIER 0 RESPONSE ADDRESS
0127 61		=2049	DB LOW(DDAMEM) ; DATA TYPE MODIFIER 1 RESPONSE ADDRESS
0128 63		=2050	DB LOW(DRM) ; DATA TYPE MODIFIER 2 RESPONSE ADDRESS

LOC	OBJ	LINE	SOURCE STATEMENT
0129	69	=2051	DB LOW(DINTRG) ; DATA TYPE MODIFIER 3 RESPONSE ADDRESS
012A	65	=2052	DB LOW(DPRBRK) ; DATA TYPE MODIFIER 4 RESPONSE ADDRESS
012B	67	=2053	DB LOW(DDABRK) ; DATA TYPE MODIFIER 5 RESPONSE ADDRESS
002C		=2054	STRGOC EQU LOW \$
012C	6B	=2055	DB LOW(DNOBRK) ; EXECUTION MODE MODIFIER 0
012D	6D	=2056	DB LOW(DMBRK) ; EXECUTION MODE MODIFIER 1
012E	6F	=2057	DB LOW(DSS) ; EXECUTION MODE MODIFIER 2
012F	72	=2058	DB LOW(DPA) ; EXECUTION MODE MODIFIER 3
0130	75	=2059	DB LOW(DTR) ; EXECUTION MODE MODIFIER 4
		=2060 ;	
		=2061 ;	UTILITY OUTPUT MESSAGES
		=2062 ;	
		=2063	DEKRROR:
0131	79	=2064	DB 01111001D ; "E"
0132	50	=2065	DB 01010000B ; "R"
0133	50	=2066	DB 01010000B ; "R"
0134	5C	=2067	DB 01011100B ; "O"
0135	50	=2068	DB 01010000B ; "R"
0136	00	=2069	DB 11000000B ; "-."
		=2070	DSGNON:
0137	00	=2071	DB 00000000B ; " "
0138	76	=2072	DB 01110110B ; "H"
0139	6D	=2073	DB 01101101B ; "S"
013A	79	=2074	DB 01111001B ; "E"
013B	40	=2075	DB 01000000B ; "-"
013C	66	=2076	DB 01100110B ; "4"
013D	E7	=2077	DB 11100111B ; "9."(TM)
		=2078	DRUN:
013E	00	=2079	DB 00000000B ; " "
013F	40	=2080	DB 01000000B ; "-"
0140	50	=2081	DB 01010000B ; "R"
0141	1C	=2082	DB 00011100B ; "U"
0142	54	=2083	DB 01010100B ; "N"
0143	00	=2084	DB 11000000B ; "-."
		=2085	DEFINT:
0144	73	=2086	DB 01110011B ; "P"
0145	B9	=2087	DB 10111001B ; "C."
		=2088	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=2089 ;	
		=2090 ;	PRIMARY COMMAND RESPONSE STRING PATTERNS
		=2091 ;	
		=2092 DMOD:	
0146	79	=2093	DB 01111001B, 00111001B, 11110100B ; "ECH. "
0147	39	=	
0148	F4	=	
		=2094 DGO:	
0149	3D	=2095	DB 00111101B, 11011100B ; "GO. "
014A	DC	=	
		=2096 DFILL:	
014B	71	=2097	DB 01110001B, 00110000B, 10111000B ; "FIL. "
014C	30	=	
014D	B8	=	
		=2098 DLST:	
014E	38	=2099	DB 00111000B, 01101101B, 11111000B ; "LST. "
014F	6D	=	
0150	F8	=	
		=2100 DREC:	
0151	3E	=2101	DB 00111110B, 01110011B, 10111000B ; "UPL. "
0152	73	=	
0153	B8	=	
		=2102 DREL:	
0154	5E	=2103	DB 01011110B, 01010100B, 10111000B ; "DNL. "
0155	54	=	
0156	B8	=	
		=2104 DSB:	
0157	6D	=2105	DB 01101101B, 01111000B, 11111100B ; "STB. "
0158	78	=	
0159	FC	=	
		=2106 DCB:	
015A	39	=2107	DB 00111001B, 00111000B, 11111100B ; "CLB. "
015B	38	=	
015C	FC	=	
		=2108 DGR:	
015D	3D	=2109	DB 00111101B, 11010000B ; "GR. "
015E	D0	=	
		=2110 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		=2111 ;	
		=2112 ;	MEMORY SPACE MODIFIER OPTION RESPONSE STRINGS:
		=2113 ;	
		=2114 DFRMEM:	
015F	73	=2115	DB 01110011B, 11010000B ; "PR. "
0160	D0	=	
		=2116 DDAMEM:	
0161	5E	=2117	DB 01011110B, 11110111B ; "DA. "
0162	F7	=	
		=2118 DRM:	
0163	50	=2119	DB 01010000B, 10111101B ; "RG. "
0164	BD	=	
		=2120 DFRBRK:	
0165	73	=2121	DB 01110011B, 11111100B ; "PB. "
0166	FC	=	
		=2122 DDBRBRK:	
0167	5E	=2123	DB 01011110B, 11111100B ; "DB. "
0168	FC	=	
		=2124 DINTRG:	
0169	76	=2125	DB 01110110B, 11010000B ; "HR. "
016A	D0	=	
		=2126 ;	
		=2127 ;	RESPONSE MESSAGES FOR GU CONDITION MODIFIERS.
		=2128 ;	
		=2129 DNOBRK:	
016B	54	=2130	DB 01010100B, 11111100B ; "NB. "
016C	FC	=	
		=2131 DMBRK:	
016D	7C	=2132	DB 01111100B, 11010000B ; "BR. "
016E	D0	=	
		=2133 DSS:	
016F	6D	=2134	DB 01101101B, 01101101B, 11111000B ; "SST. "
0170	6D	=	
0171	F8	=	
		=2135 DPA:	
0172	77	=2136	DB 01110111B, 01111100B, 11010000B ; "ABR. "
0173	7C	=	
0174	D0	=	
		=2137 DTR:	
0175	77	=2138	DB 01110111B, 01101101B, 11111000B ; "AST. "
0176	6D	=	
0177	F8	=	
		=2139 ;	
		=2140	SIZECHK
0078		=2143+	SIZE SET 120
		=2144+;	
		=2145+;	*****
		=2154	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=2155	CODEEOLK 45
00C0		=2160+	ORG 192
		=2164 ;	INPADR INPUT DATA INTO TWO-BYTE PARAMETER BUFFER INDICATED BY R0.
		=2165 ;	RECEIVE NUMERIC KEYS FROM KEYBOARD UNTIL ',' OR '.'.
		=2166 ;	SHIFT INTO ADDRESS BUFFER;
		=2167 ;	RE-WRITE DISPLAY.
		=2168 ;	IF NUMBER OF CONSTANTS NEEDED IS ZERO, NO NEW PARAMETERS ARE ALLOWED.
		=2169 ;	
00C0 97		=2170 INPADR:	CLR C
00C1 R7		=2171	CPL C
		=2172	MNOV A, NUMCON
00C2 B938		=2181+	MOV R1, #NUMCON
00C4 F1		=2182+	MOV A, @R1
00C5 C6D7		=2186	JZ ELSIF1
00C7 FB		=2187 INPAD1:	MOV A, KEY
00C8 92D7		=2188	JBA ELSIF1
00CA 20		=2189	XCH A, @R0
00CB 47		=2190	SWAP A
00CC 20		=2191	XCH A, @R0
00CD 30		=2192	XCHD A, @R0
00CE 18		=2193	INC R0
00CF 30		=2194	XCHD A, @R0
00D0 3478		=2195	CALL UPDADR
00D2 14EC		=2196	CALL INPKEY
00D4 97		=2197	CLR C
00D5 04C7		=2198	JMP INPAD1
		=2199 ;	
		=2200 ;	ELSIF1 IF KEY=', ' OR '.' THEN RETURN.
		=2201 ;	
00D7 FB		=2202 ELSIF1:	MOV A, KEY
00D8 D312		=2203	XRL A, #KEYNXT
00DA C6E5		=2204	JZ ELSIF2
00DC FB		=2205	MOV A, KEY
00DD D313		=2206	XRL A, #KEYEND
00DF C6E5		=2207	JZ ELSIF2
		=2208 ;	
		=2209 ;	ELSE GOTO PERROR.
		=2210 ;	
00E1 B802		=2211	MOV LDATA, #2
00E3 249A		=2212	JMP PERROR
00E5 B846		=2213 ELSIF2:	MOV R0, #SEGMAP
00E7 B903		=2214	MOV R1, #3
00E9 B4F5		=2215	CALL DELANK
00EB 83		=2216	RET
		=2217	SIZECHK
002C		=2220+ SIZE SET 44	
		=2221+;	
		=2222+;	*****
		=2231	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=2232	CODEBLK 35
0178		=2242+	ORG 376
		=2246	;UPDADR UPDATE ADDRESS FIELD
		=2247 ;	(LAST THREE CHARACTERS OF DISPLAY) WITH ADDRESS BUFFER
		=2248	UPDADR: MMOV NEXTPL, PLUS3
0178	D93A	=2259+	MOV R1, #NEXTPL
017A	D103	=2260+	MOV @R1, #PLUS3
		=2264 ;	WRITE ADDR INTO NEXT THREE BUFFER LOCATIONS.
017C	F0	=2265	UPDADR1: MOV A, @R0
017D	C8	=2266	DEC R0
017E	530F	=2267	ANL A, #0FH
0180	960E	=2268	JNZ DSPHI
0182	D408	=2269	CALL WDISP
0184	F0	=2270	MOV A, @R0
0185	47	=2271	SWAP A
0186	530F	=2272	ANL A, #0FH
0188	9692	=2273	JNZ DSPM1
018A	D408	=2274	CALL WDISP
018C	2494	=2275	JMP DSPLO
018E	D403	=2276	DSPHI: CALL DSPACC
0190	F0	=2277	DSPMID: MOV A, @R0
0191	47	=2278	SWAP A
0192	D403	=2279	DSPM1: CALL DSPACC
0194	F0	=2280	DSPLO: MOV A, @R0
0195	D403	=2281	CALL DSPACC
0197	83	=2282	RET
		=2283	SIZECHK
0020		=2286+	SIZE SET 32
		=2287+;	
		=2288+;	*****
		=2297	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=2298	CODEBLK 35
0198		=2308+	ORG 400
		=2312 ;	PERROR: REPEAT
		=2313 ;	OUTPUT_MESSAGE(PERROR_PROMPT)
		=2314 ;	OUTPUT(LDATA)
		=2315 ;	CALL INPUT_BYTE(KEY)
		=2316 ;	UNTIL KEY='CLEAR/PREVIOUS'
0198 B804		=2317 PERROR:	MOV LDATA, #4
019A BF02		=2318 PERROR:	MOV XPCODE, #2
019C 74D1		=2319	CALL XPTTEST
019E 27		=2320	CLR A
019F D7		=2321	MOV PSW, A
01A0 FB		=2322	MOV A, KEY
01A1 D317		=2323	XRL A, #KEYCLR
01A3 C6D6		=2324	JZ ERROR2
01A5 27		=2325	CLR A
01A6 3400		=2326	CALL OUTUTL
01A8 FA		=2327	MOV A, LDATA
01A9 D4D3		=2328	CALL DSPACC
		=2329	MMOV KBDBUF, NEG1
01AB B93B		=2340+	MOV R1, #KBDBUF
01AD B1FF		=2341+	MOV @R1, #NEG1
01AF 14EC		=2345	CALL INPKEY
01B1 FB		=2346	MOV A, KEY
01B2 D313		=2347	XRL A, #KEYEND
01B4 9698		=2348	JNZ ERROR
01B6 0429		=2349 ERROR2:	JMP MAIN
		=2350	SIZECHK
0020		=2353+	SIZE SET 32
		=2354+;	
		=2355+;	*****
		=2364 ;	
		=2365	CODEBLK 80
0200		=2380+	ORG 512
		=2384 ;	IMPLEM IMPLEMENT COMMAND
0200 2306		=2385 IMPLEM:	MOV A, #LOW(JMPTBL)
		=2386	MADD A, BCODE
0202 B936		=2392+	MOV R1, #BCODE
0204 61		=2393+	ADD A, @R1
0205 B3		=2397	JMPP @A
		=2398 ;	
		=2399 JMPTBL:	
0206 0F		=2400	DB LOW(JTOMOD)
0207 20		=2401	DB LOW(JTOGO)
0208 22		=2402	DB LOW(JTOFIL)
0209 1R		=2403	DB LOW(JTOLST)
020A 11		=2404	DB LOW(JTOREC)
020B 16		=2405	DB LOW(JTOREL)
020C 2C		=2406	DB LOW(COMSBR)
020D 28		=2407	DB LOW(COMCBR)
020E 26		=2408	DB LOW(JGORES)
		=2409 ;	
020F 444F		=2410 JTOMOD:	JMP EXAMIN
		=2411 ;	
0211 85		=2412 JTOREC:	CLR F0 ; F0=0 ==> HEX FORMAT DATA DUMP

LOC	OBJ	LINE	SOURCE STATEMENT
0212	0472	=2413	CALL HFILED
0214	0429	=2414	JMP MAIN
		=2415 ;	
0216	5497	=2416	JTOREL: CALL HRECIN
0218	0429	=2417	JMP MAIN
		=2418 ;	
021A	05	=2419	JTOLST: CLR F0
021B	95	=2420	CPL F0
021C	0472	=2421	CALL HFILED
021E	0429	=2422	JMP MAIN
		=2423 ;	
0220	0400	=2424	JTOGO: JMP EPRUN
		=2425 ;	
0222	54E5	=2426	JTOFIL: CALL COMFIL
0224	0429	=2427	JMP MAIN
		=2428 ;	
0226	0461	=2429	JGORE: JMP COMGOR
		=2430 ;	
		=2431 ;	COMCBR COMMAND TO CLEAR BREAKPOINTS
0228	0A00	=2432	COMCBR: MOV LDATA, #0
022A	442E	=2433	JMP BRKFIL
		=2434 ;	
		=2435 ;	COMSBR COMMAND TO SET BREAKPOINTS
022C	0A01	=2436	COMSBR: MOV LDATA, #1
022E	2304	=2437	BRKFIL: MOV A, #4
		=2438	MADD TYPE, A
0230	0937	=2440+	MOV R1, #TYPE
0232	61	=2449+	ADD A, @R1
0233	A1	=2455+	MOV @R1, A
0234	F400	=2459	BRKNXT: CALL LSTORE
0236	FB	=2460	MOV A, KEY
0237	D313	=2461	XRL A, #KEYEND
0239	C64D	=2462	JZ BRKEND
023B	14EC	=2463	CALL INPKEY
		=2464	MMOV NUMCON, PLUS1
023D	0938	=2475+	MOV R1, #NUMCON
023F	0101	=2476+	MOV @R1, #PLUS1
0241	0830	=2480	MOV R0, #SMALO
0243	0000	=2481	MOV @R0, #0
		=2482	MMOV SMAHI, ZERO
0245	0931	=2493+	MOV R1, #SMAHI
0247	0100	=2494+	MOV @R1, #ZERO
0249	14C0	=2498	CALL INPADR
024B	E634	=2499	JNC BRKNXT
024D	0429	=2500	BRKEND: JMP MAIN
		=2501	SIZECHK
004F		=2504+	SIZE SET 79
		=2505+;	
		=2506+;	*****
		=2515	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=2516	CODEBLK 75
024F		=2531+	ORG 591
		=2535 ;	EXAMIN EXAMINE/MODIFY MEMORY COMMAND.
		=2536 ;	DISPLAYS MEMORY ADDRESS SPACE OPTION, ADDRESS VALUE, AND CURRENT DATA.
		=2537 ;	READS KEYBOARD AND INTERPRETS RESPONSE.
		=2538 ;	
		=2539 ;	OUTPUT_MESSAGE(<MEMORY_SPACE_OPTION><SMA>'='<DATA_BYTE>)
024F 85		=2540 EXAMIN:	CLR F0
		=2541 EXAM0:	MMOV A, TYPE
0250 B937		=2550+	MOV R1, #TYPE
0252 F1		=2551+	MOV A, R1
0253 0326		=2555	ADD A, #STRMEM ; OFFSET FOR FIRST MEMORY TYPE STRING
0255 3402		=2556	CALL OUTCLR
0257 0831		=2557	MOV R0, #SMA10+1
0259 347C		=2558	CALL UPDAD1
025B 2348		=2559	MOV A, #01001000B ; '='
025D 0408		=2560	CALL MDISP
025F 14FC		=2561	CALL LFETCH
0261 FA		=2562	MOV A, LDATA
0262 47		=2563	SWAP A
0263 04D3		=2564	CALL DSPACC
0265 FA		=2565	MOV A, LDATA
0266 04D3		=2566	CALL DSPACC
		=2567 ;	
		=2568 ;	
		=2569 ;	INPUT_KEY(KEY)
		=2570 ;	IF (KEY IS NOT NUMERIC)
		=2571 ;	IF (KEY=KEYEND) GO TO PARSER
		=2572 ;	ELSEIF (KEY=KEYNEXT)
		=2573 ;	INCREMENT <SMA>
		=2574 ;	GOTO EXAMIN
		=2575 ;	ELSEIF (KEY=KEYPREVIOUS)
		=2576 ;	DECREMENT <SMA>
		=2577 ;	GOTO EXAMIN
		=2578 ;	ELSE GOTO PERROR
		=2579 ;	
0260 14EC		=2580	CALL INPKEY
		=2581	MMOV A, KEY
026A FB		=2597+	MOV A, KEY
026B 927B		=2601	JB4 EXAM1
		=2602 ;	
		=2603 ;	APPEND DATA WITH <LOWNIB.<KEY>>
		=2604 ;	CALL LSTORE
		=2605 ;	GOTO EXAMIN
		=2606 ;	
026D FA		=2607	MOV A, LDATA
026E 47		=2608	SWAP A
026F 53F0		=2609	ANL A, #0F0H
0271 B675		=2610	JF0 EXAM5
0273 27		=2611	CLR A
0274 95		=2612	CPL F0
0275 6B		=2613 EXAM5:	ADD A, KEY
0276 AA		=2614	MOV LDATA, A
0277 F400		=2615	CALL LSTORE
0279 4450		=2616	JMP EXAM0

LOC	OBJ	LINE	SOURCE STATEMENT
		=2617 ;	
027B	D313	=2618 EXAM1:	XRL A, #(KEYEND)
027D	9681	=2619	JNZ EXAM2
027F	0429	=2620	JMP MAIN
		=2621 ;	
0281	FB	=2622 EXAM2:	MOV A, KEY
0282	D312	=2623	XRL A, #KEYNXT
0284	968A	=2624	JNZ EXAM3
0286	34F2	=2625	CALL INCSMA
0288	444F	=2626	JMP EXAMIN
028A	FB	=2627 EXAM3:	MOV A, KEY
028B	D317	=2628	XRL A, #KEYCLR
028D	9693	=2629	JNZ EXAM4
028F	54F4	=2630	CALL DECSMA
0291	444F	=2631	JMP EXAMIN
0293	B003	=2632 EXAM4:	MOV LDATA, #03H
0295	249A	=2633	JMP PERROR
		=2634	SIZECHK
0048		=2637+ SIZE SET 72	
		=2638+;	
		=2639+; *****	
		=2648 ;	
		=2649	CODEBLK 4
00CC		=2654+ ORG 236	
00EC	D4C2	=2658 INPKY:	CALL KBDIN ; RETURNS KEY DEPRESSION IN A
00EE	AD	=2659	MOV KEY, A
00EF	83	=2660	RET
		=2661	SIZECHK
0004		=2664+ SIZE SET 4	
		=2665+;	
		=2666+; *****	
		=2675 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		2676 \$	INCLUDE(:F0:GOCOMS.MOD)
		=2677	CODEBLK 210
0400		=2697+	ORG 1024
		=2701 ;	EPRUN RUN EMULATION MODE.
		=2702 ;	RELOAD EP WITH SYSTEM STATUS AND RELEASE.
		=2703 ;	SEQUENCE IS AS FOLLOWS:
		=2704 ;	IF COMMAND WAS TERMINATED BY THE 'NEXT' KEY:
		=2705 ;	STORE SMA INTO EP PC;
		=2706 ;	STORE EP PC INTO TOP-OF-STACK (RELATIVE TO EP PSW);
		=2707 ;	PASS EP R0;
		=2708 ;	PASS EP PSW;
		=2709 ;	PASS EP TIMER;
		=2710 ;	PASS EP ACCUMULATOR;
		=2711 ;	
0400	2302	=2712 EPRUN:	MOV R, #2
0402	3400	=2713	CALL OUTUTL
		=2714	MMOV R, NUMCON
0404	B938	=2723+	MOV R1, #NUMCON
0406	F1	=2724+	MOV R, @R1
0407	9615	=2728	JNZ EPCONT
		=2729	MMOV EPPCLO, SMAILO
0409	B930	=2745+	MOV R1, #SMAILO
040B	F1	=2746+	MOV R, @R1
040C	B924	=2752+	MOV R1, #EPPCLO
040E	R1	=2753+	MOV @R1, R
		=2756	MMOV EPPCHI, SMAHI
040F	B931	=2772+	MOV R1, #SMAHI
0411	F1	=2773+	MOV R, @R1
0412	B925	=2779+	MOV R1, #EPPCHI
0414	R1	=2780+	MOV @R1, R
0415	FB	=2783 EPCONT:	MOV R, KEY
0416	D312	=2784	XRL R, #KEYNXT
0418	C61F	=2785	JZ EPCON1
041A	2301	=2786	MOV R, #01H ; STACK ONE LEVEL DEEP TO HOLD USER STARTING ADDRESS
		=2787	MMOV EPPSW, R
041C	B921	=2800+	MOV R1, #EPPSW
041E	R1	=2801+	MOV @R1, R
		=2805 EPCON1:	MMOV LDATA, EPPCLO
041F	B924	=2821+	MOV R1, #EPPCLO
0421	F1	=2822+	MOV R, @R1
0422	RA	=2835+	MOV LDATA, R
		=2838	MMOV R, EPPSW
0423	B921	=2847+	MOV R1, #EPPSW
0425	F1	=2848+	MOV R, @R1
0426	07	=2852	DEC R
0427	5307	=2853	ANL R, #07H
0429	E7	=2854	RL R
042H	0308	=2855	ADD R, #00H
		=2856	MMOV SMAILO, R
042C	B930	=2869+	MOV R1, #SMAILO
042E	R1	=2870+	MOV @R1, R
042F	F4C3	=2874	CALL EPSTOR
		=2875	MINC SMAILO
0431	B930	=2880+	MOV R1, #SMAILO
0433	F1	=2881+	MOV R, @R1

LOC	OBJ	LINE	SOURCE STATEMENT
0434	17	=2885+	INC R
0435	R1	=2890+	MOV @R1, R
		=2893	MMOV R, EPPSW
0436	B921	=2902+	MOV R1, #EPPSW
0438	F1	=2903+	MOV R, @R1
0439	53F0	=2907	ANL R, #0F0H
		=2908	MORL R, EPPCHI
043B	B925	=2914+	MOV R1, #EPPCHI
043D	41	=2915+	ORL R, @R1
043E	0A	=2919	MOV LDATH, R
043F	F4C3	=2920	CALL EPSTOR
0441	B8D1	=2921 EPCNT:	MOV R0, #LOW(OV2BAS+OV5SIZE)
0443	746A	=2922	CALL OVLOAD
		=2923	MMOV R, EPR0
0445	B923	=2932+	MOV R1, #EPR0
0447	F1	=2933+	MOV R, @R1
0448	F4D0	=2937	CALL EPPRSS
		=2938	MMOV R, EPPSW
044A	B921	=2947+	MOV R1, #EPPSW
044C	F1	=2948+	MOV R, @R1
044D	F4D0	=2952	CALL EPPRSS
		=2953	MMOV R, EPTMR
044F	B922	=2962+	MOV R1, #EPTMR
0451	F1	=2963+	MOV R, @R1
0452	F4D0	=2967	CALL EPPRSS
		=2968	MMOV R, EPACC
0454	B920	=2977+	MOV R1, #EPACC
0456	F1	=2978+	MOV R, @R1
0457	F4D0	=2982	CALL EPPRSS
0459	8903	=2983	ORL P1, #00000011B
045B	F4D0	=2984	CALL EPSTEP
045D	745A	=2985	CALL OVSWAP
045F	046B	=2986	JMP CGO
		=2987 ;	
		=2988 ;	COMGOR GO FROM RESET COMMAND
		=2989 ;	RESET PROCESSOR
		=2990 ;	RELOAD LOW ORDER PROGRAM BYTES INTO PROGRAM MEMORY
		=2991 ;	
0461	2302	=2992 COMGOR:	MOV R, #2
0463	3400	=2993	CALL OUTUTL
0465	8910	=2994	ORL P1, #EPRSET
0467	745A	=2995	CALL OVSWAP
0469	99EF	=2996	ANL P1, #(NOT EPRSET)
		=2997 ;	
		=2998 ;	
		=2999 ;	CGO SET UP BREAK LOGIC FOR APPROPRIATE BREAK CONDITIONS,
		=3000 ;	DEPENDING ON CONTENTS OF 'TYPE'.
		=3001 ;	
		=3002 CGO:	MMOV R, TYPE
046B	B937	=3011+	MOV R1, #TYPE
046D	F1	=3012+	MOV R, @R1
046E	0371	=3016	ADD R, #LOW GOTBL
0470	B3	=3017	JMPP @R1
		=3018 ;	
0471	7C	=3019 GOTBL:	DB LOW(CGONE)

LOC	OBJ	LINE	SOURCE STATEMENT
0472	76	=3020	DB LOW(CGOMB)
0473	80	=3021	DB LOW(CGOSS)
0474	76	=3022	DB LOW(CGOPAT)
0475	80	=3023	DB LOW(CGOTRA)
		=3024 ;	
		=3025	CGOPAT:
0476	99FD	=3026	CGOMB: ANL P1, #NOT 00000010B
0478	8901	=3027	ORL P1, #0000001B
047A	8482	=3028	JMP EPRUN4
		=3029 ;	
047C	99FC	=3030	CGOMB: ANL P1, #NOT 00000011B
047E	8482	=3031	JMP EPRUN4
		=3032 ;	
		=3033	CGOTRA:
0480	8903	=3034	CGOSS: ORL P1, #00000011B
		=3035 ;	
		=3036	;EPRUN4 SET UP CONTROL LOGIC TO RUN USER'S PROGRAM.
		=3037 ;	RELEASE PROCESSOR TO RUN.
		=3038 ;	
0482	8A20	=3039	EPRUN4: ORL P2, #00100000B ;DISABLE EP LINK REFERENCES.
0484	90EF	=3040	ANL P2, #NOT 00010000B ;SET ALL REFERENCES TO RAM ARRAY.
0486	99DF	=3041	ANL P1, #NOT MODOUT
0488	F4F4	=3042	CALL EPREL
		=3043 ;	
		=3044 ;	WAIT FOR KEYSTROKE INPUT OR HARDWARE BREAK TO OCCUR.
		=3045 ;	
048A	F4BC	=3046	EPRUN1: CALL TOPPOL
048C	F4BF	=3047	CALL KBDPOL
048E	37	=3048	CPL A
048F	F295	=3049	JB7 EPRUN3
0491	8699	=3050	JNI EPRUN2
0493	848A	=3051	JMP EPRUN1
		=3052 ;	
		=3053	;EPRUN3 A KEYSTROKE WAS DETECTED WHILE EP WAS RUNNING.
		=3054 ;	BREAK EXECUTION.
		=3055 ;	PROCESS KEYSTROKE.
0495	B400	=3056	EPRUN3: CALL STSAVE
0497	84B3	=3057	JMP EPRUN5
		=3058 ;	
		=3059	;EPRUN2 AN ENABLED BREAK CONDITION OCCURRED.
		=3060 ;	BREAK EMULATION MODE,
		=3061 ;	CONTINUE ACCORDING TO GO COMMAND TYPE.
0499	B400	=3062	EPRUN2: CALL STSAVE
		=3063	MMOV A, TYPE
049B	B937	=3072+	MOV R1, #TYPE
049D	F1	=3073+	MOV A, @R1
049E	03A1	=3077	ADD A, #LOW CNTTBL
04A0	B3	=3078	JMPP @A
		=3079 ;	
04A1	A6	=3080	CNTTBL: DB LOW(BRKERR)
04A2	BA	=3081	DB LOW(EPRUN6)
04A3	BA	=3082	DB LOW(EPRUN6)
04A4	AA	=3083	DB LOW(CNTTRA)
04A5	AA	=3084	DB LOW(CNTTRA)
		=3085 ;	

LOC	OBJ	LINE	SOURCE STATEMENT
		=3086 ;	BRKERR BREAKPOINT LATCH WAS SET THOUGH BREAKPOINTS NOT ENABLED.
		=3087 ;	DISPLAY HARDWARE ERROR MESSAGE.
04A6	BAB0	=3088 BRKERR:	MOV LDATA, #00H
04A8	249A	=3089	JMP PERROR
		=3090 ;	
		=3091 CNTTRA:	MMOV R, DSPTIM
04AA	D928	=3100+	MOV R1, #DSPTIM
04AC	F1	=3101+	MOV R, R1
04AD	94F2	=3105	CALL DELAY
04AF	F4AF	=3106	CALL KBOPOL
04B1	F241	=3107	JB7 EPCNT ; B7 SET INDICATES NO KEYSTROKE.
		=3108 ;	
		=3109 ;	EPRUN5 INPUT(KEY),
		=3110 ;	IF KEY=END GO TO PARSER.
		=3111 ;	INPUT KEY,
		=3112 ;	IF KEYONEXT GO TO PARSER.
		=3113 ;	CONTINUE IN SAME MODE.
		=3114 ;	
04B3	14EC	=3115 EPRUN5:	CALL INPKEY
04B5	FB	=3116	MOV R, KEY
04B6	D313	=3117	XRL R, #KEYEND
04B8	96C7	=3118	JNZ EPRET
04BA	14EC	=3119 EPRUN6:	CALL INPKEY
04BC	FB	=3120	MOV R, KEY
04BD	D312	=3121	XRL R, #KEYNXT
04BF	96C7	=3122	JNZ LPRET
04C1	2302	=3123	MOV R, #2
04C3	3400	=3124	CALL OUTUTL
04C5	8441	=3125	JMP EPCNT
		=3126 ;	
		=3127 ;	EPRET EXECUTION MODE IS TO BE TERMINATED.
		=3128 ;	JUMP INTO PARSER TO INTERPRET KEY ALREADY DETECTED.
04C7	0433	=3129 EPRET:	JMP MAIN2
		=3130 ;	
		=3131	SIZECHK
00C9		=3134+	SIZE SET 201
		=3135+;	
		=3136+;	*****
		=3145	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=3146	CODEBLK 115
0500		=3171+	ORG 1200
		=3175 ;	STSAVE EP STATUS SAVE SUBROUTINE.
		=3176 ;	FORCE CALL TO LOC 014H;
		=3177 ;	SAVE EP ACC;
		=3178 ;	SAVE EP TIMER;
		=3179 ;	SAVE EP PSW;
		=3180 ;	SAVE EP R0;
		=3181 ;	SAVE EP TOP-OF-STACK IN EP PC;
		=3182 ;	RETURN.
0500	744F	=3183	STSAVE: CALL EPBRK
0502	2303	=3184	MOV R, #3
0504	3400	=3185	CALL OUTUTL
0506	7450	=3186	CALL OVSWNP
0508	080F	=3187	MOV R0, #LOW(OV0BAS+OVSIZE)
050A	746A	=3188	CALL OVLOAD
050C	0A20	=3189	ORL P2, #00100000B
050E	2314	=3190	MOV A, #14H
0510	91	=3191	MOVX @R1, A
0511	9ADF	=3192	ANL P2, #NOT 00100000B
0513	8903	=3193	ORL P1, #00000011B
0515	F40B	=3194	CALL EPSTEP
0517	0A20	=3195	ORL P2, #00100000B
0519	9AEF	=3196	ANL P2, #NOT 00010000B
051B	8903	=3197	ORL P1, #(ENGRAM OR ENBLNK)
051D	F40B	=3198	CALL EPSTEP
		=3199 ;	
		=3200 ;	EXECUTION PROCESSOR IS NOW AT LOCATION 009H INTERNAL WITH
		=3201 ;	(RETURN ADDRESS+2) PUSHED ON STACK.
		=3202 ;	
051F	0805	=3203	MOV R0, #LOW(OV3BAS+OVSIZE)
0521	746A	=3204	CALL OVLOAD
0523	F400	=3205	CALL EPPASS
		=3206	MMOV EPACC, A
0525	0920	=3219+	MOV R1, #EPACC
0527	A1	=3220+	MOV @R1, A
0528	F400	=3224	CALL EPPASS
		=3225	MMOV EPTMR, A
052A	0922	=3238+	MOV R1, #EPTMR
052C	A1	=3239+	MOV @R1, A
052D	F400	=3243	CALL EPPASS
		=3244	MMOV EPPSW, A
052F	0921	=3257+	MOV R1, #EPPSW
0531	A1	=3258+	MOV @R1, A
0532	F400	=3262	CALL EPPASS
		=3263	MMOV EPR0, A
0534	0923	=3276+	MOV R1, #EPR0
0536	A1	=3277+	MOV @R1, A
0537	080B	=3281	MOV R0, #LOW(OV1BAS+OVSIZE)
0539	746A	=3282	CALL OVLOAD
		=3283	MMOV A, EPPSW
053B	0921	=3292+	MOV R1, #EPPSW
053D	F1	=3293+	MOV A, @R1
053E	07	=3297	DEC A
053F	5307	=3298	ANL A, #07H

LOC	OBJ	LINE	SOURCE STATEMENT
0541	E7	=3299	RL A
0542	0300	=3300	ADD A, #00H
		=3301	MMOV SMALO, A
0544	B930	=3314+	MOV R1, #SMALO
0546	A1	=3315+	MOV @R1, A
0547	F4B7	=3319	CALL EPFET
0549	03FE	=3320	ADD A, #-2
054B	AA	=3321	MOV LDATA, A
		=3322	MMOV EPPCLO, A
054C	B924	=3335+	MOV R1, #EPPCLO
054E	A1	=3336+	MOV @R1, A
054F	F4C3	=3340	CALL EPSTOR
0551	B930	=3341	MOV R1, #SMALO
0553	11	=3342	INC @R1
0554	F4B7	=3343	CALL EPFET
0556	AA	=3344	MOV LDATA, A
0557	53F0	=3345	ANL A, #11110000B
0559	2A	=3346	XCH A, LDATA
055A	13FF	=3347	ADDC A, #-1
055C	530F	=3348	ANL A, #00001111B
		=3349	MMOV EPPCHI, A
055E	B925	=3362+	MOV R1, #EPPCHI
0560	A1	=3363+	MOV @R1, A
0561	4A	=3367	ORL A, LDATA
0562	AA	=3368	MOV LDATA, A
0563	F4C3	=3369	CALL EPSTOR
0565	B825	=3370	MOV R0, #EPPCHI
0567	347C	=3371	CALL UPDAD1
0569	2340	=3372	MOV A, #01000000B ; "-" FOR DISPLAY
056B	D408	=3373	CALL WDISP
056D	B820	=3374	MOV R0, #EPACC
056F	3490	=3375	CALL DSPMID
0571	03	=3376	RET
		=3377	SIZECHK
0072		=3380+	SIZE SET 114
		=3381+;	
		=3382+;	*****
		=3391	#EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		3392 \$	INCLUDE(:F0:HFILE.MOD)
0000		=3393	CHARCR EQU 0DH ; <CR>
000A		=3394	CHARLF EQU 0AH ; <LF>
001A		=3395	CNTRLZ EQU 1BH ; CONTROL-Z
		=3396 ;	
		=3397	CODEBLK 88
0297		=3412+	ORG 663
		=3416	;HRECIN HEXFILE RECORD INPUT ROUTINE
0297 34CD		=3417	HRECIN: CALL CHARIN
0299 D31A		=3418	XRL A, #CNTRLZ
029B C6E0		=3419	JZ DONE
029D D31A		=3420	XRL A, #CNTRLZ
029F D33A		=3421	XRL A, #'(')
02A1 9697		=3422	JNZ HRECIN
		=3423	MNOV CHKSUM, ZERO
02A3 BD00		=3428+	MOV CHKSUM, #ZERO
02A5 14F0		=3432	CALL BYTEIN
		=3433	MNOV BUF CNT, A
02A7 B941		=3446+	MOV R1, #BUF CNT
02A9 A1		=3447+	MOV @R1, A
02AA 14F0		=3451	CALL BYTEIN
		=3452	MNOV SMARHI, A
02AC B931		=3465+	MOV R1, #SMARHI
02AE A1		=3466+	MOV @R1, A
02AF 14F0		=3470	CALL BYTEIN
		=3471	MNOV SMARLO, A
02B1 B930		=3484+	MOV R1, #SMARLO
02B3 A1		=3485+	MOV @R1, A
02B4 14F0		=3489	CALL BYTEIN
		=3490	MNOV RECTYP, A
02B6 B942		=3503+	MOV R1, #RECTYP
02B8 A1		=3504+	MOV @R1, A
		=3508 ;	
		=3509	;HDATIN HEX DATA BYTE IN
		=3510	HDATIN: MNOV A, BUF CNT
02B9 B941		=3519+	MOV R1, #BUF CNT
02BB F1		=3520+	MOV A, @R1
02BC C6CC		=3524	JZ RECDON
02BE 14F0		=3525	CALL BYTEIN
02C0 AA		=3526	MOV LDATA, A
02C1 F400		=3527	CALL LSTORE
02C3 34F2		=3528	CALL INCSMA
		=3529	MDEC BUF CNT
02C5 B941		=3534+	MOV R1, #BUF CNT
02C7 F1		=3535+	MOV A, @R1
02C8 07		=3539+	DEC A
02C9 A1		=3544+	MOV @R1, A
02CA 44B9		=3547	JMP HDATIN
		=3548 ;	
02CC 34CD		=3549	RECDON: CALL CHARIN
02CE D33F		=3550	XRL A, #'(')
02D0 C6DB		=3551	JZ CKSMOK
02D2 D33F		=3552	XRL A, #'(') ; SWITCH BACK TO DATA CHARACTER
02D4 34BA		=3553	CALL NIBIN2 ; JOIN SUBROUTINE ALREADY IN PROGRESS
02D6 14F2		=3554	CALL BYTEI1 ; DITTO

LOC	OBJ	LINE	SOURCE STATEMENT
		=3555	; (RESULT FOR NON-'?' CHARACTERS IS AS IF
		=3556	; BYTEIN WAS CALLED.)
		=3557	MMOV A, CHKSUM
02D8	FD	=3573+	MOV A, CHKSUM
02D9	96E1	=3577	JNZ CHKERR
		=3578	CKSMOK: MMUV A, RECTYP
02D8	B942	=3587+	MOV R1, #RECTYP
02DD	F1	=3588+	MOV A, @R1
02DE	C697	=3592	JZ HRECIN
		=3593 ;	
		=3594 ; DONE	HEX FILE CORRECTLY RECEIVED
02E0	83	=3595 DONE:	RET
		=3596 ;	
		=3597 ;	CHKERR CHECKSUM ERROR IN INPUT RECORD DETECTED
02E1	BA0C	=3598 CHKERR:	MOV LDATA, #0CH
02E3	249A	=3599	JMP PERROR
		=3600	SIZECHK
004E		=3603+	SIZE SET 78
		=3604+;	
		=3605+;	*****
		=3614 ;	
		=3615	CODEBLK 12
00F0		=3620+	ORG 240
		=3624 ;	BYTEIN BYTE INPUT SUBROUTINE
		=3625 ;	RECEIVES TWO HEXIDECIMAL CHARACTERS FROM THE TAPE INPUT DEVICE
		=3626 ;	AND ASSEMBLES THEM INTO A SINGLE BYTE OF DATA.
00F0	34B8	=3627	BYTEIN: CALL NIBIN
00F2	47	=3628	BYTEI1: SWAP A
00F3	AA	=3629	MOV LDATA, A
00F4	34B8	=3630	CALL NIBIN
		=3631	MORL LDATA, A
00F6	4A	=3640+	ORL A, LDATA
00F7	AA	=3660+	MOV LDATA, A
00F8	6D	=3664	ADD A, CHKSUM
00F9	AD	=3665	MOV CHKSUM, A
00FA	FA	=3666	MOV A, LDATA
00FB	83	=3667	RET
		=3668	SIZECHK
006C		=3671+	SIZE SET 12
		=3672+;	
		=3673+;	*****
		=3682 ;	
		=3683	CODEBLK 25
01B8		=3693+	ORG 440
		=3697 ;	NIBIN RECEIVES A HEXIDECIMAL CHARACTER AND PRODUCES A MASKED FOUR BIT VALUE.
		=3698 ;	NOTE- ERROR CHECKING DONE TO VERIFY HEXIDECIMAL VALIDITY
01B8	34C0	=3699	NIBIN: CALL CHARIN
01BA	03C6	=3700	NIBIN2: ADD A, #-3AH ; ACC=0F6-0FF FOR CHARACTERS '0'-'9'
		=3701	; CHARACTERS > '9' PRODUCE OVERFLOW
01BC	E6C2	=3702	JNC NIBI3
01BE	03F9	=3703	ADD A, #-7 ; ACC=0-5 FOR CHARACTERS 'A'-'F'
01C0	E6C9	=3704	JNC ASCERR ; ERROR IF CHARACTER BETWEEN '9' AND 'A'
		=3705 ;	
		=3706 ;	ACC=0F01-05H FOR CHARACTERS '0'-'F'
		=3707 ;	

LOC	OBJ	LINE	SOURCE STATEMENT
01C2	03FA	=3708	NIBI3: ADD A, #6 ; ACC=0F0H-0FFH FOR CHARACTERS '0'-'F'
01C4	0310	=3709	ADD A, #10H ; ACC=00H-0FH FOR CHARACTERS '0'-'F';
		=3710	; OVERFLOW IF ABOVE IS TRUE
01C6	E6C9	=3711	JNC ASCERR
01C8	83	=3712	RET
		=3713 ;	
		=3714	;ASCERR ILLEGAL HEXIDECIMAL CHARACTER RECEIVED
01C9	BA0A	=3715	ASCERR: MOV LDATA, #0AH
01CB	249A	=3716	JMP PERROR
		=3717	SIZECHK
0015		=3720+	SIZE SET 21
		=3721+;	
		=3722+;	*****
		=3731 ;	
		=3732 ;	
		=3733	CODEBLK 5
01CD		=3743+	ORG 461
		=3747	;CHARIN CHARACTER INPUT ROUTINE.
		=3748 ;	RECEIVES ONE ASCII CHARACTER FROM THE LOGICAL READER DEVICE.
01CD	D449	=3749	CHARIN: CALL CIN
01CF	537F	=3750	ANL A, #7FH
01D1	83	=3751	RET
		=3752	SIZECHK
0005		=3755+	SIZE SET 5
		=3756+;	
		=3757+;	*****
		=3766 ;	
		=3767 ;	
		=3768	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=3769	CODEBLK 100
0572		=3794+	ORG 1394
		=3798	;HFILED HEX FILE OUTPUT SUBROUTINE
		=3799	; WHEN CALLED WITH F0=0 OUTPUT IS STANDARD HEX FILE FORMAT.
		=3800	; WHEN CALLED WITH F0=1 OUTPUT IS FORMATTED DATA DUMP TO CRT.
		=3801	HFILED: MMOV MEMHI, SMAHI
0572 B931		=3817+	MOV R1, #SMAHI
0574 F1		=3818+	MOV R1, @R1
0575 B935		=3824+	MOV R1, #MEMHI
0577 R1		=3825+	MOV @R1, A
		=3828	MMOV MEMLO, SMALO
0578 B938		=3844+	MOV R1, #SMALO
057A F1		=3845+	MOV R1, @R1
057B B934		=3851+	MOV R1, #MEMLO
057D R1		=3852+	MOV @R1, A
		=3855	MMOV CHKSUM, ZERO
057E B000		=3860+	MOV CHKSUM, #ZERO
0580 B865		=3864	MOV R0, #HEXBUF
		=3865	;
		=3866	;LDBYTE LOAD NEXT BYTE FROM MEMORY INTO HEX BUFFER
0582 14FC		=3867	LDBYTE: CALL LFETCH
0584 FA		=3868	MOV A, LDATA
0585 A0		=3869	MOV @R0, A
0586 18		=3870	INC R0
0587 B4E2		=3871	CALL CMPMRS
0589 E69C		=3872	JNC ENDFIL
058B 34F2		=3873	CALL INCSMA
058D F8		=3874	MOV A, R0
058E 038B		=3875	ADD A, #- (BUFLen+HEXBUF)
0590 E682		=3876	JNC LDBYTE
0592 D400		=3877	CALL HREC0
0594 A472		=3878	JMP HFILED
		=3879	;
		=3880	;ENDFIL END HEX FILE TRANSMISSION:
		=3881	; PRINT OUT BUFFER FOR LAST DATA RECORD
		=3882	; PRINT OUT CANNED 'END-OF-FILE' RECORD
		=3883	; RETURN
0596 D400		=3884	ENDFIL: CALL HREC0
0598 B6A7		=3885	JF0 HFDONE
059A 34D2		=3886	CALL TCRLF0
059C B8AE		=3887	MOV R0, #(<LOW EOFREC>)
059E F8		=3888	ENDF1: MOV A, R0
059F A3		=3889	MOV @R0, A
05A0 C6A7		=3890	JZ HFDONE
05A2 B4BD		=3891	CALL CHAR0
05A4 18		=3892	INC R0
05A5 A49E		=3893	JMP ENDF1
05A7 34D2		=3894	HFDONE: CALL TCRLF0
05A9 231A		=3895	MOV A, #CNTRLZ
05AB B4BD		=3896	CALL CHAR0
05AD 83		=3897	RET
		=3898	;
		=3899	;EOFREC CHARACTER SKIPPING FOR CANNED END-OF-FILE RECORD FOR
		=3900	; INTEL HEX FILE FORMAT STANDARD.
05AE 203A3830		=3901	EOFREC: DB ' :00000001FF'

LOC	OBJ	LINE	SOURCE STATEMENT
05B2	30303030		
05B6	30314646		
05BA	00	=3902	DB 0 ; END OF STRING CODE BYTE
		=3903	SIZECHK
0049		=3906+	SIZE SET 73
		=3907+	
		=3908+	*****
		=3917 ;	
		=3918 ;	
		=3919	CODEBLK 90
0600		=3949+	ORG 1536
		=3953 ;	HRECO HEXIDEcimal RECORD OUTPUT SEQUENCE.
		=3954 ;	HEX BUFFER ALREADY LOADED.
0600	F8	=3955	HRECO: MOV A, R0
0601	039B	=3956	ADD A, #-HEXBUF
		=3957	MMOV BUFcnt, A
0603	B941	=3970+	MOV R1, #BUFcnt
0605	A1	=3971+	MOV @R1, A
0606	34D2	=3975	CALL TCRlFO
0608	2320	=3976	MOV A, #' '
060A	B4BD	=3977	CALL CHAR0
060C	B617	=3978	JF0 FDUMP1
060E	233A	=3979	MOV A, #' '
0610	B4BD	=3980	CALL CHAR0
		=3981	MMOV A, BUFcnt
0612	B941	=3990+	MOV R1, #BUFcnt
0614	F1	=3991+	MOV A, @R1
0615	34DB	=3995	CALL BYTE0
		=3996	FDUMP1: MMOV A, MEMHI
0617	B935	=4005+	MOV R1, #MEMHI
0619	F1	=4006+	MOV A, @R1
061A	34DB	=4010	CALL BYTE0
		=4011	MMOV A, MEMLO
061C	B934	=4020+	MOV R1, #MEMLO
061E	F1	=4021+	MOV A, @R1
061F	34DB	=4025	CALL BYTE0
0621	B620	=4026	JF0 FDUMP2
0623	27	=4027	CLR A
0624	34DB	=4028	CALL BYTE0
0626	C42C	=4029	JMP DAT0
0628	233D	=4030	FDUMP2: MOV A, #'='
062A	B4BD	=4031	CALL CHAR0
		=4032 ;	DAT0 DATA OUTPUT
062C	B865	=4033	DAT0: MOV R0, #HEXBUF
062E	B632	=4034	DAT01: JF0 FDUMP5
0630	C436	=4035	JMP FDUMP3
0632	2320	=4036	FDUMP5: MOV A, #' '
0634	B4BD	=4037	CALL CHAR0
0636	F0	=4038	FDUMP3: MOV A, @R0
0637	34DB	=4039	CALL BYTE0
0639	18	=4040	INC R0
		=4041	MOJNZ BUFcnt, DAT01
063A	B941	=4046+	MOV R1, #BUFcnt
063C	F1	=4047+	MOV A, @R1
063D	07	=4051+	DEC A

LOC	OBJ	LINE	SOURCE STATEMENT
063E	R1	=4056+	MOV BRL, A
063F	962E	=4060+	JNZ DAT01
		=4062 ;	
		=4063 ;	ENDREC END RECORD BEING TRANSMITTED
0641	B648	=4064	ENDREC: JF0 FDUMP4
		=4065	MMOV A, CHKSUM
0643	FD	=4081+	MOV A, CHKSUM
0644	37	=4085	CPL A
0645	17	=4086	INC A
0646	34D8	=4087	CALL BYTE0
0648	83	=4088	FDUMP4: RET
		=4089	SIZECHK
0049		=4092+	SIZE SET 73
		=4093+;	
		=4094+;	*****
		=4103 ;	
		=4104	CODEBLK 9
01D2		=4114+	ORG 466
		=4118 ;	TCRLF0 TAPE <CR><LF> OUTPUT
01D2	230D	=4119	TCRLF0: MOV A, #CHARCR
01D4	B4BD	=4120	CALL CHAR0
01D6	230H	=4121	MOV A, #CHARLF
01D8	B4BD	=4122	CALL CHAR0
01DA	83	=4123	RET
		=4124	SIZECHK
0009		=4127+	SIZE SET 9
		=4128+;	
		=4129+;	*****
		=4138 ;	
		=4139	CODEBLK 11
01D8		=4149+	ORG 475
		=4153 ;	BYTE0 BYTE OUTPUT
01D8	FA	=4154	BYTE0: MOV LDATA, A
01DC	6D	=4155	ADD A, CHKSUM
01DD	AD	=4156	MOV CHKSUM, A
01DE	FA	=4157	MOV A, LDATA
01DF	47	=4158	SWAP A
01E0	B4BB	=4159	CALL NIB0
01E2	FA	=4160	MOV A, LDATA
01E3	B4BB	=4161	CALL NIB0
01E5	83	=4162	RET
		=4163	SIZECHK
0008		=4166+	SIZE SET 11
		=4167+;	
		=4168+;	*****
		=4177 ;	
		=4178	CODEBLK 12
01E6		=4188+	ORG 486
		=4192 ;	HEXASC HEXIDECIMAL NIBBLE TO ASCII CHARACTER CONVERSION.
01E6	530F	=4193	HEXASC: ANL A, #0FH
01E8	03F6	=4194	ADD A, #(-10)
01EA	F6EF	=4195	JC HEXNIB
01EC	033A	=4196	ADD A, #(<10+'0')
01EE	83	=4197	RET
01EF	0341	=4198	HEXNIB: ADD A, #('A')

LOC	OBJ	LINE	SOURCE STATEMENT
01F1	83	=4199	RET
		=4200	SIZECHK
000C		=4203+	SIZE SET 12
		=4204+;	
		=4205+;	*****
		=4214 ;	
		=4215 ;	
		=4216	DECLARE BITSO, CONST
000B		=4230	BITSO EQU 11 ; DATA BITS PUT OUT (INCLUDING TWO STOP BITS)
		=4231 ;	
		=4232	CODEBLK 30
04C9		=4252+	ORG 1225
		=4256	; HBDLAY HALF-BIT TIME DELAY
		=4257	HBDLAY: MMOV H, HBITHI
04C9 B927		=4273+	MOV R1, HBITHI
04CB F1		=4274+	MOV A, @R1
04CC B945		=4280+	MOV R1, #H
04CE R1		=4281+	MOV @R1, A
		=4284	MMOV R1, HBITLO
04CF B926		=4300+	MOV R1, HBITLO
04D1 F1		=4301+	MOV A, @R1
04D2 B9		=4314+	MOV R1, A
04D3 04D7		=4317	JMP HBD1
04D5 B900		=4318	HBD2: MOV R1, #0
04D7 E9D7		=4319	HBD1: DJNZ R1, HBD1
		=4320	MDJNZ H, HBD2
04D9 B945		=4325+	MOV R1, #H
04DB F1		=4326+	MOV A, @R1
04DC 07		=4330+	DEC A
04DD R1		=4335+	MOV @R1, A
04DE 96D5		=4339+	JNZ HBD2
04E0 83		=4341	RET
		=4342	SIZECHK
0018		=4345+	SIZE SET 24
		=4346+;	
		=4347+;	*****
		=4356 ;	
		=4357	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=4358	CODEBLK 46
05B8		=4383+	ORG 1467
		=4387 ;NIB0	MASK ACC TO MAKE HEX NIBBLE, TRANSLATE TO ASCII AND OUTPUT
05B8 34E6		=4388 NIB0:	CALL HEXASC
		=4389 ;	
		=4390 ;CHAR0	CONSOLE OUTPUT SUBROUTINE
		=4391 ;	WRITES THE CONTENTS OF THE ACC TO THE CRT DISPLAY SCREEN
		=4392 CHAR0:	MMOV REGC, A
05B8 B944		=4405+	MOV R1, #REGC
05BF A1		=4406+	MOV @R1, A
		=4410	MMOV B, BITS0 ;SET NUMBER OF BITS TO BE TRANSMITTED
05C0 B943		=4421+	MOV R1, #B
05C2 B10B		=4422+	MOV @R1, #BITS0
05C4 97		=4426	CLR C ;CLEAR CARRY
05C5 F6CB		=4427 C01:	JC C02
05C7 99BF		=4428	ANL P1, #NOT TTYOUT
05C9 A4CF		=4429	JMP C03
05CB 8940		=4430 C02:	ORL P1, #TTYOUT
05CD 00		=4431	NOP ;EVEN OUT TWO BRANCH EXECUTION TIMES
05CE 00		=4432	NOP
05CF 94C9		=4433 C03:	CALL HBDLAY
05D1 94C9		=4434	CALL HBDLAY
05D3 97		=4435	CLR C ;SET WHAT WILL EVENTUALLY BECOME A STOP BIT
05D4 A7		=4436	CPL C
		=4437	RRRC REGC ;ROTATE CHARACTER RIGHT ONE BIT,
05D5 B944		=4442+	MOV R1, #REGC
05D7 F1		=4443+	MOV A, @R1
05D8 67		=4447+	RRC A
05D9 A1		=4452+	MOV @R1, A
		=4455	; \ MOVING NEXT DATA BIT INTO CARRY
		=4456	MOJNZ B, C01 ;CHECK IF CHARACTER (AND STOP BIT(S)) DONE
05DA B943		=4461+	MOV R1, #B
05DC F1		=4462+	MOV A, @R1
05DD 07		=4466+	DEC A
05DE A1		=4471+	MOV @R1, A
05DF 96C5		=4475+	JNZ C01
05E1 83		=4477	RET
		=4478	SIZECHK
0027		=4481+	SIZE SET 39
		=4482+;	
		=4483+;	*****
		=4492 ;	
		=4493	CODEBLK 47
0649		=4523+	ORG 1609
		=4527 ;CIN	CONSOLE INPUT SUBROUTINE WAITS FOR A KEYSTROKE AND
		=4528 ;	RETURNS WITH 8 BITS IN REG ACC.
0649 B943		=4529 CIN:	MOV R1, #B
064B B10B		=4530	MOV @R1, #0 ;DATA BITS TO BE READ
064D 464D		=4531 C10:	JNT1 C10
064F 464D		=4532	JNT1 C10
0651 5651		=4533 C11:	JT1 C11
0653 5651		=4534	JT1 C11
0655 94C9		=4535	CALL HBDLAY
0657 5651		=4536	JT1 C11
0659 94C9		=4537 C12:	CALL HBDLAY

LOC	OBJ	LINE	SOURCE STATEMENT
065B	94C9	=4538	CALL HBDLAY
065D	5662	=4539	JT1 C13 ; CHECK SID LINE LEVEL
065F	97	=4540	CLR C ; DATA BIT IN CY
0660	C465	=4541	JMP C14
0662	97	=4542 C13:	CLR C
0663	A7	=4543	CPL C
0664	00	=4544	NOP ; EVEN OUT BRANCH EXECUTION TIMES
0665	00	=4545 C14:	NOP
0666	00	=4546	NOP
0667	00	=4547	NOP
		=4548	MRRC REGC
0668	B944	=4553+	MOV R1, #REGC
066A	F1	=4554+	MOV A, @R1
066B	67	=4558+	RRC A
066C	A1	=4563+	MOV @R1, A
		=4566	MOJNZ B, C12
066D	B943	=4571+	MOV R1, #B
066F	F1	=4572+	MOV A, @R1
0670	07	=4576+	DEC A
0671	A1	=4581+	MOV @R1, A
0672	9659	=4585+	JNZ C12
		=4587	MMOV A, REGC
0674	B944	=4596+	MOV R1, #REGC
0676	F1	=4597+	MOV A, @R1
0677	83	=4601	RET ; CHARACTER COMPLETE
		=4602	SIZECHK
062F		=4605+	SIZE SET 47
		=4606+	
		=4607+	*****
		=4616	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		4617 \$	INCLUDE(:F0:MEMREF.MOD)
		=4618	CODEBLK 15
02E5		=4633+	ORG 741
		=4637 ;	CONFIL COMMAND TO FILL ADDRESS SPACE BETWEEN SMA AND EMA WITH DATA
		=4638 ;	IN LOW BYTE OF MEM.
		=4639	CONFIL: MMOV LDATA, MEMLO
02E5 B934		=4655+	MOV R1, #MEMLO
02E7 F1		=4656+	MOV A, @R1
02E8 AA		=4669+	MOV LDATA, A
02E9 F400		=4672	LFILL: CALL LSTORE
02EB B4E2		=4673	CALL CNPMAS
02ED E6F3		=4674	JNC LFILL1
02EF 34F2		=4675	CALL INCSMA
02F1 44E9		=4676	JMP LFILL
02F3 83		=4677	LFILL1: RET
		=4678	SIZECHK
000F		=4681+	SIZE SET 15
		=4682+;	
		=4683+;	*****
		=4692 ;	
		=4693	CODEBLK 4
00FC		=4698+	ORG 252
		=4702 ;	LFETCH FETCHES CONTENTS OF LOGICAL MEMORY ADDRESS DETERMINED BY
		=4703 ;	<TYPE>, <SMAHI>, & <SMALO> INTO <LDATA>.
00FC D478		=4704	LFETCH: CALL AFETCH
00FE AA		=4705	MOV LDATA, A
00FF 83		=4706	RET
		=4707	SIZECHK
0004		=4710+	SIZE SET 4
		=4711+;	
		=4712+;	*****
		=4721 ;	
		=4722	CODEBLK 75
0678		=4752+	ORG 1656
		=4756 ;	
		=4757 ;	AFETCH LOGICAL FETCH SUBROUTINE
		=4758 ;	FETCHS CONTENTS OF VARIOUS MEMORY SPACES TO ACC.
		=4759	AFETCH: MMOV A, TYPE
0678 B937		=4768+	MOV R1, #TYPE
067A F1		=4769+	MOV A, @R1
067B 037E		=4773	ADD A, #LOW LFETBL
067D B3		=4774	JMPP @A
		=4775 ;	
067E 04		=4776	LFETBL: DB LOW LFEPM
067F 98		=4777	DB LOW LFEDM
0680 9C		=4778	DB LOW LFEREG
0681 A9		=4779	DB LOW LFEINT
0682 B1		=4780	DB LOW LFEPRK
0683 B1		=4781	DB LOW LFEPRK
		=4782 ;	
		=4783	LFEPM: MMOV A, SMAHI
0684 B931		=4792+	MOV R1, #SMAHI
0686 F1		=4793+	MOV A, @R1
0687 9698		=4797	JNZ LFEDM
		=4798	MMOV A, SMALO

LOC	OBJ	LINE	SOURCE STATEMENT
0689	B930	=4807+	MOV R1, #SMALO
068B	F1	=4808+	MOV A, @R1
068C	03E9	=4812	ADD A, #-OVSZIE
068E	F698	=4813	JC LFEDM
		=4814	MMOV A, SMALO
0690	B930	=4823+	MOV R1, #SMALO
0692	F1	=4824+	MOV A, @R1
0693	034E	=4820	ADD A, #OYBUF
0695	A9	=4829	MOV R1, A
0696	F1	=4830	MOV A, @R1
0697	83	=4831	RET
0698	94E1	=4832	LFEDM: CALL LPGSEL
069A	81	=4833	MOVX A, @R1
069B	83	=4834	RET
		=4835 ;	
		=4836	LFEREC: MMOV A, SMALO
069C	B930	=4845+	MOV R1, #SMALO
069E	F1	=4846+	MOV A, @R1
069F	537F	=4850	ANL A, #01111111B ; CHECK IF LOW 7 BITS =0
06A1	06A5	=4851	JZ LFER0
06A3	E4B7	=4852	JMP EPFET
		=4853 ;	
		=4854	LFER0: MMOV A, EPR0
06A5	B923	=4863+	MOV R1, #EPR0
06A7	F1	=4864+	MOV A, @R1
06A8	83	=4868	RET
		=4869 ;	
		=4870	LFEINT: MMOV A, SMALO
06A9	B930	=4879+	MOV R1, #SMALO
06AB	F1	=4880+	MOV A, @R1
06AC	0320	=4884	ADD A, #EPACC
06AE	A9	=4885	MOV R1, A
06AF	F1	=4886	MOV A, @R1
06B0	83	=4887	RET
		=4888 ;	
		=4889	;LFEBRK LOGICAL FETCH OF BREAK-POINT DATA
06B1	94E1	=4890	LFEBRK: CALL LPGSEL
06B3	99F7	=4891	ANL P1, #NOT 00001000B
06B5	8908	=4892	ORL P1, #00001000B
06B7	99FD	=4893	ANL P1, #NOT 00000010B
06B9	8901	=4894	ORL P1, #00000001B
06BB	81	=4895	MOVX A, @R1
06BC	2301	=4896	MOV A, #01H
06BE	86C1	=4897	JNI LFEBR1
06C0	27	=4898	CLR A
06C1	83	=4899	LFEBR1: RET
		=4900	SIZECHK
004A		=4903+	SIZE SET 74
		=4904+;	
		=4905+;	*****
		=4914	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=4915	CODEBLK 85
0700		=4950+	ORG 1792
		=4954 ;	
		=4955 ;	LSTORE LOGICAL STORE SUBROUTINE
		=4956 ;	STORES CONTENTS OF LDATA INTO VARIOUS MEMORY SPACES.
		=4957	LSTORE: MMOV A, TYPE
0700 B937		=4966+	MOV R1, #TYPE
0702 F1		=4967+	MOV A, @R1
0703 0306		=4971	ADD A, #LOW LSTTBL
0705 B3		=4972	JMPP @A
		=4973 ;	
0706 0C		=4974	LSTTBL: DB LOW LSTPM
0707 21		=4975	DB LOW LSTDH
0708 26		=4976	DB LOW LSTREG
0709 34		=4977	DB LOW LSTINT
070A 3D		=4978	DB LOW LSTBRK
070C 3D		=4979	DB LOW LSTBRK
		=4980 ;	
		=4981	LSTPM: MMOV A, SMAHI
070C B931		=4990+	MOV R1, #SMAHI
070E F1		=4991+	MOV A, @R1
070F 9621		=4995	JNZ LSTDH
		=4996	MMOV A, SMAILO
0711 B930		=5005+	MOV R1, #SMAILO
0713 F1		=5006+	MOV A, @R1
0714 03E9		=5010	ADD A, #-OVSZ
0716 F621		=5011	JC LSTDH
		=5012	MMOV A, SMAILO
0718 B930		=5021+	MOV R1, #SMAILO
071A F1		=5022+	MOV A, @R1
071B 034E		=5026	ADD A, #OVBUF
071D A9		=5027	MOV R1, A
071E FA		=5028	MOV A, LDATA
071F A1		=5029	MOV @R1, A
0720 83		=5030	RET
		=5031 ;	
0721 94E1		=5032	LSTDH: CALL LPGSEL
0723 FA		=5033	MOV A, LDATA
0724 91		=5034	MOVX @R1, A
0725 83		=5035	RET
		=5036 ;	
		=5037	LSTREG: MMOV A, SMAILO
0726 B930		=5046+	MOV R1, #SMAILO
0728 F1		=5047+	MOV A, @R1
0729 537F		=5051	ANL A, #01111111B ; CHECK IF LOW ORDER BITS = 0
072B 062F		=5052	JZ LSTR0
072D E4C3		=5053	JMP EPSTOR
		=5054 ;	
		=5055	LSTR0: MMOV EPTR0, LDATA
072F FA		=5076+	MOV A, LDATA
0730 B923		=5084+	MOV R1, #EPTR0
0732 A1		=5085+	MOV @R1, A
0733 83		=5088	RET
		=5089 ;	
		=5090	LSTINT: MMOV A, SMAILO

LOC	OBJ	LINE	SOURCE STATEMENT
0734	B930	=5099+	MOV R1, #SMALO
0736	F1	=5100+	MOV A, @R1
0737	0320	=5104	ADD A, #EPACC
0739	A9	=5105	MOV R1, A
073A	FA	=5106	MOV A, LDATA
073B	A1	=5107	MOV @R1, A
073C	03	=5108	RET
		=5109 ;	
		=5110 ;	LSTBRK LOGICAL STORE OF BREAK-POINT DATA
073D	94E1	=5111	LSTBRK: CALL LPGSEL
073F	FA	=5112	MOV A, LDATA
0740	1246	=5113	JB0 LSTBR1
0742	8901	=5114	ORL P1, #00000001B
0744	E448	=5115	JMP LSTBR2
0746	99FE	=5116	LSTBR1: ANL P1, #NOT 00000001B
0748	99F7	=5117	LSTBR2: ANL P1, #NOT 00001000B
074A	01	=5118	MOVX A, @R1
074B	0908	=5119	ORL P1, #00001000B
074D	03	=5120	RET
		=5121	SIZECHK
004E		=5124+	SIZE SET 70
		=5125+;	
		=5126+;	*****
		=5135 ;	
		=5136	CODEBLK 17
04E1		=5156+	ORG 1249
		=5160 ;	LPGSEL LOGICAL PAGE SELECT.
		=5161 ;	SETS UP PORT 2 TO ADDRESS APPROPRIATE BYTE OF RAM BLOCK.
		=5162	LPGSEL: MOV A, TYPE
04E1	B937	=5171+	MOV R1, #TYPE
04E3	F1	=5172+	MOV A, @R1
04E4	5301	=5176	ANL A, #00000001B ; MASK OFF DATA TYPE SELECTOR BIT
04E6	47	=5177	SWAP A
		=5178	MORL A, SMANI
04E7	B931	=5184+	MOV R1, #SMANI
04E9	41	=5185+	ORL A, @R1
04EB	4340	=5189	ORL A, #01000000B
04EC	3A	=5190	OUTL P2, A
		=5191	MOV A, SMALO
04ED	B930	=5200+	MOV R1, #SMALO
04EF	F1	=5201+	MOV A, @R1
04F0	A9	=5205	MOV R1, A
04F1	03	=5206	RET
		=5207	SIZECHK
0011		=5210+	SIZE SET 17
		=5211+;	
		=5212+;	*****
		=5221 ;	
		=5222	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=5223	CODEBLK 11
01F2		=5233+	ORG 496
		=5237	; INCSMA INCREMENT STARTING MEMORY ADDRESS WORD.
01F2 B930		=5238	INCSMA: MOV R1, #SMALO
01F4 11		=5239	INCM: INC @R1
01F5 F1		=5240	MOV A, @R1
01F6 96FC		=5241	JNZ INCM1
01F8 19		=5242	INC R1
01F9 F1		=5243	MOV A, @R1
01FA 17		=5244	INC A
01FB 31		=5245	XCHD A, @R1
01FC 83		=5246	INCM1: RET
		=5247	SIZECHK
000B		=5250+	SIZE SET 11
		=5251+;	
		=5252+;	*****
		=5261	;
		=5262	CODEBLK 12
02F4		=5277+	ORG 756
		=5281	; DECSMA DECREMENT SMA WORD.
02F4 B930		=5282	DECSMA: MOV R1, #SMALO
02F6 F1		=5283	MOV A, @R1
02F7 07		=5284	DEC A
02F8 21		=5285	XCH A, @R1
02F9 96FF		=5286	JNZ DECSM1
02FB 19		=5287	INC R1
02FC F1		=5288	MOV A, @R1
02FD 07		=5289	DEC A
02FE 31		=5290	XCHD A, @R1
02FF 83		=5291	DECSM1: RET
		=5292	SIZECHK
000C		=5295+	SIZE SET 12
		=5296+;	
		=5297+;	*****
		=5306	;
		=5307	CODEBLK 15
05E2		=5332+	ORG 1506
		=5336	; CMPMAS COMPARE MEMORY ADDRESSES
		=5337	; COMPARE SMA BYTES WITH EMA BYTES TO DETERMINE RELATIVE MAGNITUDE.
		=5338	; RETURNS WITH CARRY=1 IFF <SMA> >= <EMA>.
		=5339	; IS CALLED AFTER ACTION HAS BEEN PERFORMED ON <SMA> TO DETERMINE IF
		=5340	TASK IS COMPLETED:
		=5341	; IF CY=0 THEN <SMA> >= <EMA> ==> TERMINATE TASK.
		=5342	; IF CY=1 THEN <SMA> < <EMA> ==> INC SMA AND REPEAT.
		=5343	CMPMAS: MMOV A, SMALO
05E2 B930		=5352+	MOV R1, #SMALO
05E4 F1		=5353+	MOV A, @R1
05E5 37		=5357	CPL A
		=5358	MADD A, EMALO
05E6 B932		=5364+	MOV R1, #EMALO
05E8 61		=5365+	RDD A, @R1
		=5369	MMOV A, SMAHI
05E9 B931		=5378+	MOV R1, #SMAHI
05EB F1		=5379+	MOV A, @R1
05EC 37		=5383	CPL A

LOC	OBJ	LINE	SOURCE STATEMENT
		=5384	MADDC A,EMAH1
05ED	B933	=5390+	MOV R1, #EMAH1
05EF	71	=5391+	ADDC A, @R1
05F0	83	=5395	CMPRET: RET
		=5396	SIZECHK
000F		=5399+	SIZE SET 15
		=5400+	
		=5401+	*****
		=5410	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		5411 \$	INCLUDE(:F0:KBD.MOD)
		=5412	CODEBLK 100
074E		=5447+	ORG 1870
		=5451 ;	
		=5452 ;	KEYBOARD AND DISPLAY PROCESSING ROUTINE
		=5453 ;	CALLED PERIODICALLY WHEN KBD AND DISPLAY ARE TO BE ALIVE
074E D5		=5454 TIINT:	SEL RB1
		=5455	MMOV ASAVE, A
074F B93E		=5468+	MOV R1, #ASAVE
0751 A1		=5469+	MOV @R1, A
0752 23F0		=5473	MOV A, #(-10H)
0754 62		=5474	MOV T, A ;RELOAD TIMER INTERVAL
0755 27		=5475	CLR A
0756 3E		=5476	MOVD PSEGH1, A ;WRITE BLANK PATTERN TO SEG DRIVERS
0757 3D		=5477	MOVD PSEGLO, A
0758 FD		=5478	MOV A, CURDIG
0759 07		=5479	DEC A
075A 3F		=5480	MOVD PDIGIT, A ;ENERGIZE CHARACTER
075B 0C		=5481	MOVD A, PINFUT ;LOAD ANY SWITCH CLOSURES
075C AA		=5482	MOV ROTPAT, A
		=5483	;WRITE NEXT SEGMENT PATTERN
075D FD		=5484	MOV A, CURDIG
075E 07		=5485	DEC A
075F 0346		=5486	ADD A, #SEGMAP ;ADD CURDIG DISPLACEMENT TO BASE
0761 A8		=5487	MOV R0, A
0762 F0		=5488	MOV A, @R0 ;LOAD ACC W/ NEXT SEGMENT PATTERN
0763 3D		=5489	MOVD PSEGLO, A ;ENABLE APPROPRIATE SEGMENTS
0764 47		=5490	SWAP A
0765 3E		=5491	MOVD PSEGH1, A
		=5492 ;	
		=5493 ;	*****
		=5494 ;	THE NEXT CHARACTER IS NOW BEING DISPLAYED.
		=5495 ;	THE KEYBOARD SCAN ROUTINE IS INTEGRATED INTO THE DISPLAY SCAN.
		=5496 ;	WITH THE CURRENT ROW ENERGIZED, CHECK IF THERE ARE ANY INPUTS.
		=5497 ;	*****
		=5498 ;	
		=5499 ;	ROTATE BITS THROUGH THE CY WHILE INCREMENTING KEYLOC.
		=5500 ;	
0766 BB04		=5501	MOV ROTCNT, #NCOLS ;SET UP FOR <NCOLS> LOOPS THROUGH 'NXTLOC'
		=5502 NXTLOC:	MRRC ROTPAT
0768 FA		=5514+	MOV A, ROTPAT
0769 67		=5518+	RRC A
076A AA		=5529+	MOV ROTPAT, A
076B F68B		=5532	JC SCANS ;ONE BIT IN CY INDICATES KEY NOT DOWN
076D BE01		=5533	MOV KEYFLG, #1 ;MARK THAT AT LEAST ONE KEY WAS DETECTED
		=5534	; \ IN THE CURRENT SCAN
		=5535 ;	
		=5536 ;	*****
		=5537 ;	A KEYSYROKE WAS DETECTED FOR THE CURRENT COLUMN. ITS
		=5538 ;	POSITION IS IN REGISTER KEYLOC. SEE IF SAME KEY SENSED LAST CYCLE.
		=5539 ;	*****
		=5540 ;	
		=5541	MMOV A, KEYLOC
076F B93C		=5550+	MOV R1, #KEYLOC
0771 F1		=5551+	MOV A, @R1

LOC	OBJ	LINE	SOURCE STATEMENT
0772	2C	=5555	XCH A, LASTKY
0773	DC	=5556	XRL A, LASTKY
0774	067C	=5557	JZ SCAN3
		=5558 ;	
		=5559 ;	*****
		=5560 ;	A DIFFERENT KEY WAS READ ON THIS CYCLE THAN ON THE PREVIOUS CYCLE.
		=5561 ;	SET NREPTS TO THE DEBOUNCE PARAMETER FOR A NEW COUNTDOWN.
		=5562 ;	*****
		=5563 ;	
0776	B93D	=5564	MOV R1, #NREPTS
0778	B106	=5565	MOV @R1, #6
077A	E488	=5566	JMP SCAN5
		=5567 ;	
		=5568 ;	*****
		=5569 ;	SAME KEY WAS DETECTED 3S ON PREVIOUS CYCLE
		=5570 ;	LOOK AT NREPTS: IF ALREADY ZERO, DO NOTHING.
		=5571 ;	ELSE DECREMENT NREPTS.
		=5572 ;	IF THIS RESULTS IN ZERO, MOVE LASTKY INTO KBDBUF.
		=5573 ;	*****
		=5574 ;	
		=5575	SCAN3: MMOV A, NREPTS
077C	B93D	=5584+	MOV R1, #NREPTS
077E	F1	=5585+	MOV A, @R1
077F	0688	=5589	JZ SCAN5 ; IF ALREADY ZERO
0781	07	=5590	DEC A ; INDICATE ONE MORE SUCCESSIVE KEY DETECTION
		=5591	MMOV NREPTS, A
0782	B93D	=5604+	MOV R1, #NREPTS
0784	A1	=5605+	MOV @R1, A
0785	9688	=5609	JNZ SCAN5 ; IF DECREMENT DOES NOT RESULT IN ZERO
		=5610	MMOV KBDBUF, LASTKY ; TO MARK NEW KEY CLOSURE
0787	FC	=5633+	MOV A, LASTKY
0788	B93B	=5639+	MOV R1, #KBDBUF
0788	A1	=5640+	MOV @R1, A
		=5643 ;	
0788	B93C	=5644	SCAN5: MOV R1, #KEYLOC
078D	11	=5645	INC @R1
078E	ED68	=5646	DJNZ ROTCNT, NXTLOC
0790	EDAC	=5647	DJNZ CURDIG, TIRET1
0792	BD88	=5648	MOV CURDIG, #CHARNO
		=5649 ;	
		=5650 ;	*****
		=5651 ;	THE FOLLOWING CODE SEGMENT IS USED BY THE KEYBOARD SCANNING ROUTINE.
		=5652 ;	IT IS EXECUTED ONLY AFTER A REFRESH SEQUENCE IS COMPLETED
		=5653 ;	*****
		=5654 ;	
		=5655	MMOV KEYLOC, ZERO
0794	B93C	=5666+	MOV R1, #KEYLOC
0796	B100	=5667+	MOV @R1, #ZERO
0798	FE	=5671	MOV A, KEYFLG
0799	969D	=5672	JNZ SCAN8 ; JUMP IF ANY KEYS WERE DETECTED
		=5673	MMOV LASTKY, NEG1 ; CHANGE <LASTKY> WHEN NO KEYS ARE DOWN
079B	BCFF	=5678+	MOV LASTKY, #NEG1
079D	BE00	=5682	SCAN8: MOV KEYFLG, #0
		=5683 ;	
		=5684 ;	*****

LOC	OBJ	LINE	SOURCE STATEMENT
		=5685 ;	
		=5686 ;	KBD/DISP RETURN CODE- RESTORES SYSTEM STATUS.
		=5687	MMOV A, RDELAY
079F	B93F	=5696+	MOV R1, #RDELAY
07A1	F1	=5697+	MOV A, @R1
07A2	06A8	=5701	JZ TIRET1
07A4	07	=5702	DEC A
		=5703	MMOV RDELAY, A
07A5	B93F	=5716+	MOV R1, #RDELAY
07A7	A1	=5717+	MOV @R1, A
		=5721 TIRET1:	MMOV A, RSAVE
07A8	B93E	=5730+	MOV R1, #RSAVE
07AA	F1	=5731+	MOV A, @R1
07AB	93	=5735	RETR
		=5736 ;	
		=5737 ;	
		=5738 ;	TOFPOL TIMER OVERFLOW POLLING SUBROUTINE.
		=5739 ;	CALLED REPEATEDLY FROM WHEREVER KBD/DISP MUST BE ALIVE.
		=5740 ;	MONITORS THE TIMER OVERFLOW FLAG (TOF) AND CALLS SERVICE
		=5741 ;	ROUTINE WHEN APPROPRIATE.
07AC	164E	=5742 TOFPOL:	JTF TIINT
07AE	83	=5743	RET
		=5744	SIZECHK
0061		=5747+ SIZE SET 97	
		=5748+;	
		=5749+; *****	
		=5750 \$EJECT	

LOC	OBJ	LINE	SOURCE STATEMENT
		=5759	CODEBLK 17
06C2		=5789+	ORG 1730
		=5793 ;	
		=5794 ;	KBDIN: KEYBOARD INPUT SUBROUTINE.
		=5795 ;	RETURNS ONLY AFTER A NEW KEYSTROKE HAS BEEN DETECTED AND DEBOUNCED.
		=5796 ;	VALUE OF KEY POSITION IN SWITCH MATRIX IS
		=5797 ;	RETURNED IN THE ACCUMULATOR.
		=5798 ;	DISPLAY CHARACTER NOW ON BLANKED BEFORE RETURNING.
06C2	0F03	=5799 KBDIN:	MOV XPCODE, #3
06C4	74D1	=5800	CALL XPTST
06C6	F4AC	=5801 KBDI1:	CALL TOFPOL
		=5802	MMOV A, KDBUF
06C8	B73E	=5811:	MOV R1, #KDBUF
06CA	F1	=5812:	MOV A, @R1
06CB	F2C6	=5816	JB7 KBDI1
06CD	27	=5817	CLR A
06CE	3E	=5818	MOVD PSEGH, A
06CF	1D	=5819	MOVD PSEGLO, A
06D0	37	=5820	CPL A
06D1	21	=5821	XCH A, @R1
06D2	03	=5822	RET
		=5823	SIZECHK
0011		=5826+ SIZE SET 17	
		=5827+;	
		=5828+; *****	
		=5837 ;	
		=5838	CODEBLK 15
05F1		=5863+	ORG 1521
		=5867 ;	CLEAR: WRITES 'BLANK' CHARACTERS INTO ALL DISPLAY REGISTERS.
		=5868 ;	RETURNS WITH NEXTPL SET TO LEFTMOST CHARACTER POSITION
		=5869 ;	DOES NOT AFFECT ACC OR CY.
05F1	E846	=5870 CLEAR:	MOV R0, #SEGMAP
05F3	B908	=5871	MOV R1, #CHARNO
05F5	B000	=5872 DBLANK:	MOV @R0, #0 ; STORE THE BLANK CODE
05F7	18	=5873	INC R0 ; POINT TO NEXT CHARACTER TO THE LEFT
05F9	E9F5	=5874	DJNZ R1, DBLANK
		=5875	MMOV NEXTPL, CHARNO
05FA	B93A	=5886+	MOV R1, #NEXTPL
05FC	B100	=5887+	MOV @R1, #CHARNO
05FE	83	=5891	RET
		=5892	SIZECHK
000E		=5895+ SIZE SET 14	
		=5896+;	
		=5897+; *****	
		=5906 ;	
		=5907	CODEBLK 44
06D3		=5937+	ORG 1747
		=5941 ;	DSFACC: DISPLAY VALUE OF LOW NIBBLE OF ACC
06D3	530F	=5942 DSFACC:	ANL A, #0FH
06D5	03EF	=5943	ADD A, #DGPATS
06D7	A3	=5944	MOVP A, @A
		=5945 ;	WDISP: WRITES BIT PATTERN NOW IN ACC INTO NEXT CHARACTER POSITION
		=5946 ;	OF THE DISPLAY (NEXTPL). INCREMENTS NEXTPL.
		=5947 ;	RESULTS IN DISPLAY BEING FILLED LEFT TO RIGHT, THEN RESTARTING
06D8	AE	=5948 WDISP:	MOV DSPTMP, A

LOC	OBJ	LINE	SOURCE STATEMENT
06D9	BFB4	=5949	MOV XPCODE, #4
06DB	74D1	=5950	CALL XPTST
		=5951	MMOV R, NEXTPL
06DD	B93A	=5960+	MOV R1, #NEXTPL
06DF	F1	=5961+	MOV R, @R1
06E0	0345	=5965	ADD R, #SEGMAP-1
06E2	A9	=5966	MOV R1, R
06E3	FE	=5967	MOV R, DSPTMP
06E4	A1	=5968	MOV @R1, R
		=5969	MDJNZ NEXTPL, WDISP1
06E5	B93A	=5974+	MOV R1, #NEXTPL
06E7	F1	=5975+	MOV R, @R1
06E8	07	=5979+	DEC R
06E9	A1	=5984+	MOV @R1, R
06EA	96EE	=5988+	JNZ WDISP1
06EC	B108	=5990	MOV @R1, #CHARNO
06EE	83	=5991	WDISP1: RET
		=5992 ;	
		=5993 ;	DGPATS IS THE BASE FOR THE TABLE OF SEGMENT PATTERNS FOR HEX DIGITS.
		=5994 ;	HERE THE FULL HEX SET (0-F) IS INCLUDED.
		=5995 ;	
00EF		=5996	DGPATS EQU \$ AND 0FFH
		=5997 ;	
		=5998 ;	FORMAT IS PGFEDCBA IN STANDARD SEVEN-SEGMENT ENCODING CONVENTION
		=5999 ;	WHERE P REPRESENTS THE DECIMAL POINT
06EF	3F	=6000	DB 00111111B ; SEGMENT PATTERN FOR DIGIT '0'
06F0	06	=6001	DB 00000110B ; SEGMENT PATTERN FOR DIGIT '1'
06F1	5B	=6002	DB 01011011B ; SEGMENT PATTERN FOR DIGIT '2'
06F2	4F	=6003	DB 01001111B ; SEGMENT PATTERN FOR DIGIT '3'
06F3	66	=6004	DB 01100110B ; SEGMENT PATTERN FOR DIGIT '4'
06F4	6D	=6005	DB 01101101B ; SEGMENT PATTERN FOR DIGIT '5'
06F5	7D	=6006	DB 01111101B ; SEGMENT PATTERN FOR DIGIT '6'
06F6	07	=6007	DB 00000111B ; SEGMENT PATTERN FOR DIGIT '7'
06F7	7F	=6008	DB 01111111B ; SEGMENT PATTERN FOR DIGIT '8'
06F8	67	=6009	DB 01100111B ; SEGMENT PATTERN FOR DIGIT '9'
06F9	77	=6010	DB 01110111B ; SEGMENT PATTERN FOR DIGIT 'A'
06FA	7C	=6011	DB 01111100B ; SEGMENT PATTERN FOR DIGIT 'B'
06FB	39	=6012	DB 00111001B ; SEGMENT PATTERN FOR DIGIT 'C'
06FC	5E	=6013	DB 01011110B ; SEGMENT PATTERN FOR DIGIT 'D'
06FD	79	=6014	DB 01111001B ; SEGMENT PATTERN FOR DIGIT 'E'
06FE	71	=6015	DB 01110001B ; SEGMENT PATTERN FOR DIGIT 'F'
		=6016	SIZECHK
002C		=6019+	SIZE SET 44
		=6020+	
		=6021+	*****
		=6030 ;	
		=6031	CODEBLK 12
04F2		=6051+	ORG 1266
		=6055 ;	DELAY SUBROUTINE WAITS FOR THE NUMBER OF COMPLETE
		=6056 ;	DISPLAY SCANS CORRESPONDING TO THE ACC CONTENTS.
		=6057 ;	USED WITH CRUDE HUMAN INTERFACES- AS WHEN OPERATOR SHOULD SEE
		=6058 ;	SOME DISPLAY CHANGE WHILE IT IS CHANGING.
		=6059	DELAY: MMOV RDELAY, R
04F2	B93F	=6072+	MOV R1, #RDELAY
04F4	A1	=6073+	MOV @R1, R

LOC	OBJ	LINE	SOURCE STATEMENT
04F5	F4AC	=6077	DELAY1: CALL 70FFPOL
		=6078	MMOV A, RDELAY
04F7	093F	=6087+	MOV R1, #RDELAY
04F9	F1	=6088+	MOV A, @R1
04FA	96F5	=6092	JNZ DELAY1
04FC	83	=6093	RET
		=6094	SIZECHK
0000		=6097+	SIZE SET 11
		=6098+;	
		=6099+;	*****
		=6100 ;	
		=6109	CODEBLK 8
07AF		=6144+	ORG 1967
		=6148 ;	KBDPOL POLL STATUS OF KEYBOARD INPUT ROUTINE.
		=6149 ;	RETURN WITH ACC BIT 7 = 0 IF KEYBOARD INPUT HAS BEEN RECEIVED.
07AF	BF05	=6150	KBDPOL: MOV XPCODE, #5
07B1	74D1	=6151	CALL XPTST
		=6152	MMOV A, KBDDBUF
07B3	B93B	=6161+	MOV R1, #KBDDBUF
07B5	F1	=6162+	MOV A, @R1
07B6	83	=6166	RET
		=6167	SIZECHK
0000		=6170+	SIZE SET 8
		=6171+;	
		=6172+;	*****
		=6181	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		6182 \$	INCLUDE(:F0:LINK.MOD)
		=6183	CODEBLK 15
07B7		=6218+	ORG 1975
		=6222	;EPFET FETCH DATA BYTE FROM EP INTERNAL RAM ADDRESSED BY SMALO.
		=6223	EPFET: MMOV A, SMALO
07B7 B930		=6232+	MOV R1, #SMALO
07B9 F1		=6233+	MOV A, @R1
07BA F4D0		=6237	CALL EPPASS
07BC 2380		=6238	MOV A, #10000000B
07BE F4D0		=6239	CALL EPPASS
07C0 F4D0		=6240	CALL EPPASS
07C2 83		=6241	RET
		=6242	SIZECHK
000C		=6245+	SIZE SET 12
		=6246+;	
		=6247+;	*****
		=6256 ;	
		=6257	CODEBLK 15
07C3		=6292+	ORG 1987
		=6296	;EPSTOR STORE DATA IN LDATA IN EP INTERNAL RAM AT <SMALO>
07C3 FA		=6297	EPSTOR: MOV A, LDATA
07C4 F4D0		=6298	CALL EPPASS
		=6299	MMOV A, SMALO
07C6 B930		=6308+	MOV R1, #SMALO
07C8 F1		=6309+	MOV A, @R1
07C9 537F		=6313	ANL A, #01111111B
07CB F4D0		=6314	CALL EPPASS
07CD F4D0		=6315	CALL EPPASS
07CF 83		=6316	RET
		=6317	SIZECHK
000D		=6320+	SIZE SET 13
		=6321+;	
		=6322+;	*****
		=6331	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6332 ;	THE FOLLOWING UTILITIES INVOLVE INTERCHANGES BETWEEN THE MP AND EP.
		=6333 ;	
		=6334	CODEBLK 11
07D0		=6369+	ORG 2000
		=6373 ;	EPPASS PASSES A SINGLE PARAMETER BYTE TO THE EP THROUGH THE LINK.
		=6374 ;	WRITE THE CONTENTS OF THE ACC TO THE LINK;
		=6375 ;	RELEASE THE EP;
		=6376 ;	READ THE LINK INTO THE ACC;
		=6377 ;	RETURN.
07D0 0A30		=6378 EPPASS:	ORL P2, #00110000B ; ENABLE LINK WRITES.
07D2 91		=6379	MOVX @R1, A ; WRITE ACC TO LINK.
07D3 99FE		=6380	ANL P1, #NOT ENBRM ; DISABLE BREAKPOINTS.
07D5 8902		=6381	ORL P1, #ENBLNK ; SET TO BREAK ON LINK REFERENCE.
07D7 F4DB		=6382	CALL EPSTEP
07D9 81		=6383	MOVX A, @R1
07DA 83		=6384	RET
		=6385	SIZECHK
0008		=6388+ SIZE SET 11	
		=6389+;	
		=6390+;	*****
		=6399 ;	
		=6400	CODEBLK 25
07DB		=6435+	ORG 2011
		=6439 ;	EPSTEP RELEASES EP TO RUN IN PRESENT MODE UNTIL AN ANTICIPATED
		=6440 ;	HARDWARE BREAK OCCURS.
		=6441 ;	(DUE TO SINGLE STEPPING, LINK OP- CODE FETCH, OR LINK DATA FETCH.)
		=6442 ;	MUST OCCUR WITHIN A FINITE NUMBER OF CYCLES (<40 MP CYCLES)
		=6443 ;	OR WATCHDOG TIMER WILL ASSUME A COMMUNICATIONS ERROR
		=6444 ;	BETWEEN THE MP AND EP.
07DB F4F4		=6445 EPSTEP:	CALL EPREL
07DD B90A		=6446	MOV R1, #10
07DF 86F1		=6447 EPSTE1:	JNI EPSTE2
07E1 E9DF		=6448	DJNZ R1, EPSTE1
07E3 8910		=6449	ORL P1, #EPRSET
07E5 744F		=6450	CALL EPRBK
07E7 B80B		=6451	MOV R0, #LOW(OV1BAS+OVSZ)
07E9 746A		=6452	CALL OVLOAD
07EB 99EF		=6453	ANL P1, #NOT EPRSET
07ED B80E		=6454	MOV LDATA, #0EH
07EF 249A		=6455	JMP PERROR
07F1 744F		=6456 EPSTE2:	CALL EPRBK
07F3 83		=6457	RET
		=6458	SIZECHK
0019		=6461+ SIZE SET 25	
		=6462+;	
		=6463+;	*****
		=6472 ;	
		=6473 ;	
		=6474	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6475	CODEBLK 9
07F4		=6510+	ORG 2036
		=6514 ;	EPREL RELEASES EP TO RUN IN PRESENT MODE.
		=6515 ;	SEQUENCE IS AS FOLLOWS:
		=6516 ;	PUT MEMORY ARRAY IN EP MODE;
		=6517 ;	RAISE /SSTEP;
		=6518 ;	RETURN
07F4 99F7		=6519 EPREL:	ANL P1, #NOT CLRBF ; CLEAR BREAK F/F.
07F6 8900		=6520	ORL P1, #CLRBFF ; RE-ENABLE BREAK F/F.
07F8 9ABF		=6521	ANL P2, #NOT 01000000B ; ENABLE EP CONTROL OF MEM ARRAY
07FA 8904		=6522	ORL P1, #00000100B ; FREE EP TO RUN UNTIL BREAK.
07FC 83		=6523	RET
		=6524	SIZECHK
0009		=6527+	SIZE SET 9
		=6528+;	
		=6529+;	*****
		=6530 ;	
		=6539 ;	
		=6540	CODEBLK 11
034F		=6500+	ORG 847
		=6504 ;	EPBRK REGAIN CONTROL OF MEMORY ARRAY FROM EP.
		=6505 ;	DROP /SSTEP;
		=6506 ;	WAIT 30 USECS. ;
		=6507 ;	PUT MEMORY ARRAY IN MP MODE;
		=6508 ;	RETURN
034F 99FB		=6509 EPBRK:	ANL P1, #NOT 00000100B ; FREEZE EMULATION PROCESSOR.
0351 8920		=6590	ORL P1, #MODOUT ; SIGNAL EP IS NOT RUNNING USER CODE.
0353 8905		=6591	MOV R1, #5
0355 E955		=6592	DJNZ R1, \$; DELAY FOR EP TO FINISH INSTRUCTION.
0357 8A40		=6593	ORL P2, #01000000B ; SEIZE CONTROL OF MEM ARRAY.
0359 83		=6594	RET
		=6595	SIZECHK
000B		=6598+	SIZE SET 11
		=6599+;	
		=6600+;	*****
		=6609 ;	
		=6610 ;	
		=6611	CODEBLK 16
035A		=6651+	ORG 850
		=6655 ;	OVSMP OVERLAY SWAP.
		=6656 ;	SWAPS BLOCK OF DATABYTES (USER'S PROGRAM) BETWEEN MP RAM & EP PM.
035A B865		=6657 OVSMP:	MOV R0, #OVBUFF+OVSZ
035C B917		=6658	MOV R1, #OVSZ
035E 2340		=6659	MOV A, #01000000B
0360 3A		=6660	OUTL P2, A
0361 C0		=6661 OVSMP1:	DEC R0
0362 C9		=6662	DEC R1
0363 81		=6663	MOVX A, @R1
0364 20		=6664	XCH A, @R0
0365 91		=6665	MOVX @R1, A
0366 F9		=6666	MOV A, R1
0367 9661		=6667	JNZ OVSMP1
0369 83		=6668	RET
		=6669	SIZECHK
0010		=6672+	SIZE SET 16

LOC	OBJ	LINE	SOURCE STATEMENT
		=6673+	
		=6674+	*****
		=6683 ;	
		=6684	CODEBLK 14
036A		=6724+	ORG 874
		=6728 ;	OVLOAD OVERLAY LOAD.
		=6729 ;	MOVES BLOCK OF DATABYTES (ASSEMBLED SOURCE) FROM PG3 TO EP PM.
		=6730 ;	TOP OF DATA BLOCK LOADED AND BLOCK LENGTH DETERMINED BY R0 AND R1.
036A 0917		=6731 OVLOAD:	MOV R1, #OVSIZE
036C 2340		=6732	MOV A, #01000000B
036E 3A		=6733	OUTL P2, A
036F 08		=6734 MML01:	DEC R0
0370 09		=6735	DEC R1
0371 F8		=6736	MOV R0, R0
0372 E3		=6737	MOVP3 A, @A
0373 91		=6738	MOVX @R1, A
0374 F9		=6739	MOV A, R1
0375 966F		=6740	JNZ MML01
0377 83		=6741	RET
		=6742	SIZECHK
000E		=6745+	SIZE SET 14
		=6746+	
		=6747+	*****
		=6756	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6757 ;	
		=6758 ;	=====
		=6759 ;	
		=6760 ;	THE REST OF THIS MODULE CONTAINS THE MINI-MONITORS WHICH OVERLAY
		=6761 ;	THE EMULATION PROCESSOR PROGRAM RAM TO GIVE THE
		=6762 ;	MASTER PROCESSOR ACCESS TO INTERNAL REGISTERS AND RAM OF THE EP.
		=6763 ;	
		=6764 ;	=====
		=6765 ;	
		=6766	DATABLK 22
0378		=6771+	ORG 388
		=6775 ;	
		=6776 ;	OV0- OVERLAY TO BREAK EP EXECUTION AND JUMP TO LOCATION 009H.
		=6777 ;	LOCATION 009H REACHED WITH TOP-OF-STACK = RETURN ADDRESS+2
		=6778 ;	DUE TO FORCED "CALL" DURING WHICH PC WAS INCREMENTED.
		=6779 ;	LOCS 003H & 007H CALL 009H TO SIMULATE SAME CONDITION
		=6780 ;	IF BREAK OCCURS DURING INTERRUPT CYCLE.
		=6781 ;	SOURCE CODE FOR MINI-MONITOR OVERLAYED OVER LOW ORDER PROGRAM RAM.
		=6782 ;	
0378		=6783	OV0BAS EQU \$
0378		=6784	ORG OV0BAS
0378 1409		=6785	CALL 009H
037A 00		=6786	NOP
		=6787 ;	
037B		=6788	ORG OV0BAS+003H
037B 1409		=6789	CALL 009H
037D 00		=6790	NOP
037E 00		=6791	NOP
		=6792 ;	
037F		=6793	ORG OV0BAS+007H
037F 1409		=6794	CALL 009H
0381 00		=6795	NOP
0382 00		=6796	NOP
0383 00		=6797	NOP
0384 00		=6798	NOP
0385 00		=6799	NOP
0386 00		=6800	NOP
0387 00		=6801	NOP
0388 00		=6802	NOP
0389 00		=6803	NOP
038A 00		=6804	NOP
038B 00		=6805	NOP
		=6806 ;	
038C		=6807	ORG OV0BAS+014H
038C 0409		=6808	JMP 009H
		=6809 ;	
		=6810	SIZECHK
0016		=6813+	SIZE SET 22
		=6814+;	
		=6815+;	*****
		=6824	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6825	DATABLK 22
038E		=6830+	ORG 910
		=6834 ;	
		=6835 ;	OV3- OVERLAY TO SAVE STATUS DATA AFTER BREAK.
		=6836 ;	ACC, TIMER/COUNTER, PSW (WITH F1), & RAM LOC 0 PASSED SEQUENTIALLY
		=6837 ;	TO MP.
		=6838 ;	SOURCE CODE FOR MINI-MONITOR OVERLAYED OVER LOW ORDER PROGRAM RAM.
		=6839 ;	
038E		=6840	OV3BAS EQU \$
038E		=6841	ORG OV3BAS
038E 0400		=6842	JMP 0004H
0390 00		=6843	NOP
		=6844 ;	
0391		=6845	ORG OV3BAS+003H
0391 83		=6846	RET
0392 00		=6847	NOP
0393 00		=6848	NOP
0394 00		=6849	NOP
		=6850 ;	
0395		=6851	ORG OV3BAS+007H
0395 83		=6852	RET
0396 00		=6853	NOP
		=6854 ;	
0397		=6855	ORG OV3BAS+009H
0397 90		=6856	MOVX @R0, A
0398 42		=6857	MOV A, T
0399 90		=6858	MOVX @R0, A
039A C7		=6859	MOV A, PSW
039B 7611		=6860	JF1 OV3B1
039D 53F7		=6861	ANL A, #11110111B
0311		=6862	OV3B1 EQU \$- (LOW OV3BAS)
039F 90		=6863	MOVX @R0, A
03A0 C5		=6864	SEL R00
03A1 F0		=6865	MOV A, R0
03A2 0409		=6866	JMP 0009H
		=6867 ;	
		=6868	SIZECHK
0016		=6871+	SIZE SET 22
		=6872+;	
		=6873+; *****	
		=6882	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6883	DATABLK 22
03A4		=6888+	ORG 932
		=6892 ;	
		=6893 ;	OV1- OVERLAY 1 TO GIVE MP ACCESS TO EP RAM LOCS. 01H-7FH.
		=6894 ;	SOURCE CODE FOR MINI-MONITOR OVERLAYED OVER LOW ORDER PROGRAM RAM.
		=6895 ;	
03A4		=6896	OV1BAS EQU \$
		=6897 ;	
03A4 040A		=6898	JMP OV1B1
03A6 00		=6899	NOP
		=6900 ;	
03A7		=6901	ORG OV1BAS+003H
03A7 83		=6902	RET
03A8 00		=6903	NOP
03A9 00		=6904	NOP
03AN 00		=6905	NOP
		=6906 ;	
03AB		=6907	ORG OV1BAS+007H
03AB 83		=6908	RET
03AC 00		=6909	NOP
		=6910 ;	
03AD		=6911	ORG OV1BAS+009H
03AD 50		=6912	MOVX R0, A
		=6913 ;	
000A		=6914	OV1B1 EQU \$-OV1BAS
		=6915 ;	
03AE 00		=6916	MOVX A, R0
03AF A8		=6917	MOV R0, A
03B0 00		=6918	MOVX A, R0
03B1 F213		=6919	JB7 OV1B2
03B3 20		=6920	XCH A, R0
03B4 A0		=6921	MOV R0, A
03B5 0409		=6922	JMP 009H
		=6923 ;	
0313		=6924	OV1B2 EQU \$-LOW OV1BAS
		=6925 ;	
03B7 F0		=6926	MOV A, R0
03B8 0409		=6927	JMP 009H
		=6928 ;	
		=6929	SIZECHK
0016		=6932+	SIZE SET 22
		=6933+;	
		=6934+;	*****
		=6943	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		=6944	DATABLK 23
03BA		=6949+	ORG 954
		=6953 ;	
		=6954 ;	OV2- OVERLAY TO RESTORE EP STATUS SAVED ON BREAK AND RESUME USER'S PROGRAM.
		=6955 ;	SOURCE CODE FOR MINI-MONITOR OVERLAYED OVER LOW ORDER PROGRAM RAM.
		=6956 ;	
03BA		=6957	OV2BAS EQU \$
03BA		=6958	ORG OV2BAS
03BA 0400		=6959	JMP 000H
03BC 00		=6960	NOP
		=6961 ;	
03BD		=6962	ORG OV2BAS+003H
03BD 83		=6963	RET
03BE 00		=6964	NOP
03BF 00		=6965	NOP
03C0 00		=6966	NOP
		=6967 ;	
03C1		=6968	ORG OV2BAS+007H
03C1 83		=6969	RET
03C2 00		=6970	NOP
		=6971 ;	
03C3		=6972	ORG OV2BAS+009H
03C3 90		=6973	MOVX R0, A
		=6974 ;	
03C4 00		=6975	MOVX A, R0
03C5 A8		=6976	MOV R0, A
03C6 00		=6977	MOVX A, R0
03C7 07		=6978	MOV PSW, A
03C8 A5		=6979	CLR F1
03C9 B5		=6980	CPL F1
03CA 7213		=6981	JB3 OV2B1
03CC A5		=6982	CLR F1
		=6983 ;	
0313		=6984	OV2B1 EQU \$-LOW OV2BAS
		=6985 ;	
03CD 00		=6986	MOVX A, R0
03CE 62		=6987	MOV T, A
03CF 00		=6988	MOVX A, R0
03D0 93		=6989	RETR
		=6990	SIZECHK
0017		=6993+	SIZE SET 23
		=6994+;	
		=6995+;	*****
		=7004	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		7005 ;	
		7006	CODEBLK 11
03D1		7046+	ORG 977
03D1 0A80		7050	XPTEST: ORL P2, #00H
03D3 0A		7051	IN A, P2
03D4 9A7F		7052	ANL P2, #(NOT 00H)
03D6 F2D9		7053	JB7 \$+3
03D8 83		7054	RET
03D9 F5		7055	SEL MB1
03DA 0400		7056	JMP 000H
		7057	SIZECHK
000B		7060+	SIZE SET 11
		7061+;	
		7062+;	*****
		7071 ;	
		7072	CODEBLK 13
03DC		7112+	ORG 900
03DC 28432931		7116	DB '(C)1979 INTEL'
03E0 39373920			
03E4 494E5445			
03E8 4C			
		7117	SIZECHK
000D		7120+	SIZE SET 13
		7121+;	
		7122+;	*****
		7131 ;	
		7132 ;	
		7133	RSOURCE
0100		7135+	PGSIZE SET ORGPG0-000H ; BYTES USED ON PAGE 0
00FD		7136+	PGSIZE SET ORGPG1-100H ; BYTES USED ON PAGE 1
0100		7137+	PGSIZE SET ORGPG2-200H ; BYTES USED ON PAGE 2
00E9		7138+	PGSIZE SET ORGPG3-300H ; BYTES USED ON PAGE 3
00FD		7139+	PGSIZE SET ORGPG4-400H ; BYTES USED ON PAGE 4
00F7		7140+	PGSIZE SET ORGPG5-500H ; BYTES USED ON PAGE 5
00FF		7141+	PGSIZE SET ORGPG6-600H ; BYTES USED ON PAGE 6
00FD		7142+	PGSIZE SET ORGPG7-700H ; BYTES USED ON PAGE 7
		7143+	\$EJECT

LOC	OBJ	LINE	SOURCE STATEMENT
		7145 ;	*****
		7146 ;	
		7147 ;	FILL ALL UNUSED MEMORY LOCATIONS WITH NOP OPCODES
		7148 ;	
		7149 ;	*****
		7150 ;	
		7151 \$GEN	
		7158 ;	
01FD		7160	ORG ORGPG1
		7161	REPT (200H - ORGPG1)
		7162	DB 0
		7163	ENDM
01FD 00		7164+	DB 0
01FE 00		7165+	DB 0
01FF 00		7166+	DB 0
		7168 ;	
		7175 ;	
03E9		7177	ORG ORGPG3
		7178	REPT (400H - ORGPG3)
		7179	DB 0
		7180	ENDM
03E9 00		7181+	DB 0
03EA 00		7182+	DB 0
03EB 00		7183+	DB 0
03EC 00		7184+	DB 0
03ED 00		7185+	DB 0
03EE 00		7186+	DB 0
03EF 00		7187+	DB 0
03F0 00		7188+	DB 0
03F1 00		7189+	DB 0
03F2 00		7190+	DB 0
03F3 00		7191+	DB 0
03F4 00		7192+	DB 0
03F5 00		7193+	DB 0
03F6 00		7194+	DB 0
03F7 00		7195+	DB 0
03F8 00		7196+	DB 0
03F9 00		7197+	DB 0
03FA 00		7198+	DB 0
03FB 00		7199+	DB 0
03FC 00		7200+	DB 0
03FD 00		7201+	DB 0
03FE 00		7202+	DB 0
03FF 00		7203+	DB 0
		7205 ;	
04FD		7207	ORG ORGPG4
		7208	REPT (500H - ORGPG4)
		7209	DB 0
		7210	ENDM
04FD 00		7211+	DB 0
04FE 00		7212+	DB 0
04FF 00		7213+	DB 0
		7215 ;	
05FF		7217	ORG ORGPG5
		7218	REPT (600H - ORGPG5)

LOC	OBJ	LINE	SOURCE STATEMENT
-		7219	DB 0
		7220	ENDM
05FF	00	7221+	DB 0
		7223 ;	
06FF		7225	ORG ORGPG6
		7226	REPT (700H - ORGPG6)
-		7227	DB 0
		7228	ENDM
06FF	00	7229+	DB 0
		7231 ;	
07FD		7233	ORG ORGPG7
		7234	REPT (800H - ORGPG7)
-		7235	DB 0
		7236	ENDM
07FD	00	7237+	DB 0
07FE	00	7238+	DB 0
07FF	00	7239+	DB 0
		7241 ;	
		7242 \$EJECT	

LOC	ORJ	LINE	SOURCE STATEMENT
		7243	END
USER SYMBOLS			
?A	0004	?ASAVE	0002
?B0R6	0007	?B0R7	0008
?B1R7	0008	?BCODE	0002
?CONST	0003	?CURDI	0001
?EPPCH	0002	?EPPCL	0002
?FORM4	0010	?FORM5	001E
?HREGC	0002	?HREGD	0002
?KEYLO	0002	?LASTK	0001
?NCOLS	0003	?NEG1	0003
?PLUS1	0003	?PLUS3	0003
?REGC	0002	?ROTCN	0001
?START	030C	?STRTM	0002
ASAVE	003E	ASCERR	01C9
BRKNXT	0234	BUFCNT	0041
CGOPAT	0476	CGOSS	0480
CHARO	058D	CHKERR	02E1
CIN	0649	CKSMOK	02D8
CNTTBL	04A1	CNTTRA	04AA
COMGOR	0461	COMSBR	022C
DBLANK	05F5	DBPNT	0144
DECSM1	02FF	DECSMA	02F4
DGR	015D	DINTRG	0169
DPRMEM	015F	DREC	0151
DSPII	018E	DSPL0	0194
DWBRK	016D	ELSIF1	0007
ENDFIL	0596	ENDREC	0641
EPFET	07B7	EPPASS	07D0
EPRSET	0010	EPRUN	0400
EPSSTP	0004	EPSTE1	07DF
EXAM1	027B	EXAM2	0201
FDUMP2	0628	FDUMP3	0636
HBD2	04D5	HBDLAY	04C9
HFDONE	05A7	HFILE0	0572
HREGC	002E	HREGF	002F
INPAD1	00C7	INPADR	00C8
JTGOO	0220	JTOLST	021A
KBDPOL	07AF	KCLRB	000C
KEYGO	001E	KEYLOC	003C
KEYREG	001B	KEYREL	0014
LFEBR1	06C1	LFEBRK	06B1
LFETCH	00FC	LFILL	02E9
LSTINT	0734	LSTORE	0700
MADD	0024	MADD0	0025
MAINC1	0075	MAIND	0093
MEMLO	0034	MERROR	008C
MRL	002E	MRLC	0031
NEXTPL	003A	NIBI3	01C2
NUMCON	0038	NXTLOC	07C8
ORGP02	0300	ORGP03	03E9
OUTUTL	0100	OY0BAS	0378
OY3BAS	038E	OYBUF	004E
PERROR	019A	PGSIZE	00FD
?B0PNT	0008	?B0R2	0003
?B1PNT	0007	?B1R2	0003
?BINOP	0022	?BITSO	0003
?DEBNC	0003	?DSPTI	0002
?EPPSW	0002	?EPR0	0002
?H	0002	?HBITH	0002
?HREGD	0002	?HREGF	0002
?LDATA	0000	?LENGT	0000
?NEXTP	0002	?NREPT	0002
?R1	0000	?RAM	0002
?ROTPA	0001	?RSAVE	0000
?TYPE	0002	?UNARY	002A
B	0043	BCODE	0036
BUFLN	0010	BYTE11	00F2
CGOTRA	0480	CGWB	0476
CHKSUM	0005	C10	064D
CLEAR	05F1	CLRBFF	0000
C01	05C5	C02	05C8
C02	05C8	C03	05CF
C03	05CF	CODEBL	0006
COMSIZ	0003	CTAB	0023
CTAB	0023	CURDIG	0005
DCB	015A	DDABRK	0167
DDABRK	0167	DDAMEM	0161
DEBRK	0131	DFILL	014B
DERROR	0131	DONE	02E0
DNBRK	016B	DSB	0157
DRUN	013E	DSGNON	0137
DSPTIM	0028	DSPTMP	0006
DSPTMP	0006	DSS	01CF
EMAIH	0033	EMALO	0032
EMALO	0032	ENBLNK	0002
EPDRK	034F	EPCNT	0441
EPPSW	0021	EPR0	0023
EPR0	0023	EPRUN4	0402
EPRUN4	0402	EPTIMR	0022
EPSTOR	07C3	ERROR2	01B6
EXAM5	0275	EXAMIN	024F
EXAMIN	024F	EXTMON	0000
EXTMON	0000	H	0045
GOTEL	0471	HEXASC	01E6
HDATIN	02B9	HEXBUF	0065
HEXASC	01E6	HLGB	002B
HREGA	002A	HREGC	002C
HRLGB	002B	HREGD	002D
INCH	01F4	INCH1	01FC
INIT	0000	JMPTEL	0206
JGOKES	0226	KBD11	06C6
KBD11	06C6	KBDIN	06C2
KEYCLR	0017	KEYEND	0013
KEYDM	0016	KEYFIL	0010
KEYVND	0012	KEYPAT	0015
KSETB	000B	KEYPM	001A
LASTKY	0004	LDATA	0002
LFEBR0	06A5	LFEREG	069C
LFEBR1	06A5	LFEREG	069C
LFEBR2	0748	LSTBRK	073D
LSTBR1	0746	LSTBR2	0748
LSTBR2	0748	LSTTBL	0706
LSTREG	0726	M0	0010
LSTTBL	0706	M1	0020
MAIN0	0052	MAINB	0069
MAINB	0069	MAINB0	009E
MELOCK	0002	MDJNZ	002D
MDEC	002C	MORL	0027
MMOV	0020	MODOUT	0020
MMXL	0028	MORL	0027
MXCH	0029	NCOLS	0004
MXCH	0029	NOVALS	0023
NIB0	05BB	NOVALS	0023
NIB1	05BB	NOVALS	0023
NIB2	05BB	NOVALS	0023
NIB3	05BB	NOVALS	0023
NIB4	05BB	NOVALS	0023
NIB5	05BB	NOVALS	0023
NIB6	05BB	NOVALS	0023
NIB7	05BB	NOVALS	0023
NIB8	05BB	NOVALS	0023
NIB9	05BB	NOVALS	0023
NIB10	05BB	NOVALS	0023
NIB11	05BB	NOVALS	0023
NIB12	05BB	NOVALS	0023
NIB13	05BB	NOVALS	0023
NIB14	05BB	NOVALS	0023
NIB15	05BB	NOVALS	0023
NIB16	05BB	NOVALS	0023
NIB17	05BB	NOVALS	0023
NIB18	05BB	NOVALS	0023
NIB19	05BB	NOVALS	0023
NIB20	05BB	NOVALS	0023
NIB21	05BB	NOVALS	0023
NIB22	05BB	NOVALS	0023
NIB23	05BB	NOVALS	0023
NIB24	05BB	NOVALS	0023
NIB25	05BB	NOVALS	0023
NIB26	05BB	NOVALS	0023
NIB27	05BB	NOVALS	0023
NIB28	05BB	NOVALS	0023
NIB29	05BB	NOVALS	0023
NIB30	05BB	NOVALS	0023
NIB31	05BB	NOVALS	0023
NIB32	05BB	NOVALS	0023
NIB33	05BB	NOVALS	0023
NIB34	05BB	NOVALS	0023
NIB35	05BB	NOVALS	0023
NIB36	05BB	NOVALS	0023
NIB37	05BB	NOVALS	0023
NIB38	05BB	NOVALS	0023
NIB39	05BB	NOVALS	0023
NIB40	05BB	NOVALS	0023
NIB41	05BB	NOVALS	0023
NIB42	05BB	NOVALS	0023
NIB43	05BB	NOVALS	0023
NIB44	05BB	NOVALS	0023
NIB45	05BB	NOVALS	0023
NIB46	05BB	NOVALS	0023
NIB47	05BB	NOVALS	0023
NIB48	05BB	NOVALS	0023
NIB49	05BB	NOVALS	0023
NIB50	05BB	NOVALS	0023
NIB51	05BB	NOVALS	0023
NIB52	05BB	NOVALS	0023
NIB53	05BB	NOVALS	0023
NIB54	05BB	NOVALS	0023
NIB55	05BB	NOVALS	0023
NIB56	05BB	NOVALS	0023
NIB57	05BB	NOVALS	0023
NIB58	05BB	NOVALS	0023
NIB59	05BB	NOVALS	0023
NIB60	05BB	NOVALS	0023
NIB61	05BB	NOVALS	0023
NIB62	05BB	NOVALS	0023
NIB63	05BB	NOVALS	0023
NIB64	05BB	NOVALS	0023
NIB65	05BB	NOVALS	0023
NIB66	05BB	NOVALS	0023
NIB67	05BB	NOVALS	0023
NIB68	05BB	NOVALS	0023
NIB69	05BB	NOVALS	0023
NIB70	05BB	NOVALS	0023
NIB71	05BB	NOVALS	0023
NIB72	05BB	NOVALS	0023
NIB73	05BB	NOVALS	0023
NIB74	05BB	NOVALS	0023
NIB75	05BB	NOVALS	0023
NIB76	05BB	NOVALS	0023
NIB77	05BB	NOVALS	0023
NIB78	05BB	NOVALS	0023
NIB79	05BB	NOVALS	0023
NIB80	05BB	NOVALS	0023
NIB81	05BB	NOVALS	0023
NIB82	05BB	NOVALS	0023
NIB83	05BB	NOVALS	0023
NIB84	05BB	NOVALS	0023
NIB85	05BB	NOVALS	0023
NIB86	05BB	NOVALS	0023
NIB87	05BB	NOVALS	0023
NIB88	05BB	NOVALS	0023
NIB89	05BB	NOVALS	0023
NIB90	05BB	NOVALS	0023
NIB91	05BB	NOVALS	0023
NIB92	05BB	NOVALS	0023
NIB93	05BB	NOVALS	0023
NIB94	05BB	NOVALS	0023
NIB95	05BB	NOVALS	0023
NIB96	05BB	NOVALS	0023
NIB97	05BB	NOVALS	0023
NIB98	05BB	NOVALS	0023
NIB99	05BB	NOVALS	0023
NIB100	05BB	NOVALS	0023
NIB101	05BB	NOVALS	0023
NIB102	05BB	NOVALS	0023
NIB103	05BB	NOVALS	0023
NIB104	05BB	NOVALS	0023
NIB105	05BB	NOVALS	0023
NIB106	05BB	NOVALS	0023
NIB107	05BB	NOVALS	0023
NIB108	05BB	NOVALS	0023
NIB109	05BB	NOVALS	0023
NIB110	05BB	NOVALS	0023
NIB111	05BB	NOVALS	0023
NIB112	05BB	NOVALS	0023
NIB113	05BB	NOVALS	0023
NIB114	05BB	NOVALS	0023
NIB115	05BB	NOVALS	0023
NIB116	05BB	NOVALS	0023
NIB117	05BB	NOVALS	0023
NIB118	05BB	NOVALS	0023
NIB119	05BB	NOVALS	0023
NIB120	05BB	NOVALS	0023
NIB121	05BB	NOVALS	0023
NIB122	05BB	NOVALS	0023
NIB123	05BB	NOVALS	0023
NIB124	05BB	NOVALS	0023
NIB125	05BB	NOVALS	0023
NIB126	05BB	NOVALS	0023
NIB127	05BB	NOVALS	0023
NIB128	05BB	NOVALS	0023
NIB129	05BB	NOVALS	0023
NIB130	05BB	NOVALS	0023
NIB131	05BB	NOVALS	0023
NIB132	05BB	NOVALS	0023
NIB133	05BB	NOVALS	0023
NIB134	05BB	NOVALS	0023
NIB135	05BB	NOVALS	0023
NIB136	05BB	NOVALS	0023
NIB137	05BB	NOVALS	0023
NIB138	05BB	NOVALS	0023
NIB139	05BB	NOVALS	0023
NIB140	05BB	NOVALS	0023
NIB141	05BB	NOVALS	0023
NIB142	05BB	NOVALS	0023
NIB143	05BB	NOVALS	0023
NIB144	05BB	NOVALS	0023
NIB145	05BB	NOVALS	0023
NIB146	05BB	NOVALS	0023
NIB147	05BB	NOVALS	0023
NIB148	05BB	NOVALS	0023
NIB149	05BB	NOVALS	0023
NIB150	05BB	NOVALS	0023
NIB151	05BB	NOVALS	0023
NIB152	05BB	NOVALS	0023
NIB153	05BB	NOVALS	0023
NIB154	05BB	NOVALS	0023
NIB155	05BB	NOVALS	0023
NIB156	05BB	NOVALS	0023
NIB157	05BB	NOVALS	0023
NIB158	05BB	NOVALS	0023
NIB159	05BB	NOVALS	0023
NIB160	05BB	NOVALS	0023
NIB161	05BB	NOVALS	0023
NIB162	05BB	NOVALS	0023
NIB163	05BB	NOVALS	0023
NIB164	05BB	NOVALS	0023
NIB165	05BB	NOVALS	0023
NIB166	05BB	NOVALS	0023
NIB167	05BB	NOVALS	0023
NIB168	05BB	NOVALS	0023
NIB169	05BB	NOVALS	0023
NIB170	05BB	NOVALS	0023
NIB171	05BB	NOVALS	0023
NIB172	05BB	NOVALS	0023
NIB173	05BB	NOVALS	0023
NIB174	05BB	NOVALS	0023
NIB175	05BB	NOVALS	0023
NIB176	05BB	NOVALS	0023
NIB177	05BB	NOVALS	0023
NIB178	05BB	NOVALS	0023
NIB179	05BB	NOVALS	0023
NIB180	05BB	NOVALS	0023
NIB181	05BB	NOVALS	0023
NIB182	05BB	NOVALS	0023
NIB183	05BB	NOVALS	0023
NIB184	05BB	NOVALS	0023
NIB185	05BB	NOVALS	0023
NIB186	05BB	NOVALS	0023
NIB187	05BB	NOVALS	0023
NIB188	05BB	NOVALS	0023
NIB189	05BB	NOVALS	0023
NIB190	05BB	NOVALS	0023
NIB191	05BB	NOVALS	0023
NIB192	05BB	NOVALS	0023
NIB193	05BB	NOVALS	0023
NIB194	05BB	NOVALS	0023
NIB195	05BB	NOVALS	0023
NIB196	05BB	NOVALS	0023
NIB197	05BB	NOVALS	0023
NIB198	05BB	NOVALS	0023
NIB199	05BB	NOVALS	0023

PSEGLO 000C	RDELAY 003F	RECDON 02CC	RECTYP 0042	KEGC 0044	REORG 0005	RERROR 0198	RINT 0011
ROTCNT 0003	ROTPAT 0002	RSOURC 0012	SCAN3 077C	SCAN5 078B	SCAN8 079D	SEGMAP 0046	SING 001A
SIZE 0000	SIZECH 0011	SMANI 0031	SMALO 0030	STRCOM 001D	STRGOC 002C	STRMEM 0026	STRTMP 0040
STRUTL 0019	STSAVE 0500	TCRLF0 01D2	TIINT 074E	TIRET1 07A8	TOFPOL 07AC	TIYOUT 0040	TYPE 0037
UPDAD1 017C	UPDADR 0178	VERSNO 0029	WBRK 0016	WDISP 06D8	WDISP1 06EE	XPCODE 0007	XPTEST 03D1
ZERO 0000							

ASSEMBLY COMPLETE, NO ERRORS

?A	105#	1614	1629	1637	1650	1658	1721	1787	1806	1818	1977	1985	1999	2177	2388	2444
	2546	2586	2719	2788	2796	2843	2857	2865	2898	2910	2928	2943	2958	2973	3007	3068
	3096	3207	3215	3226	3234	3245	3253	3264	3272	3288	3302	3310	3323	3331	3350	3358
	3434	3442	3453	3461	3472	3480	3491	3499	3515	3562	3583	3637	3958	3966	3986	4001
	4016	4070	4393	4401	4592	4764	4788	4803	4819	4841	4859	4875	4962	4986	5001	5017
	5042	5095	5167	5180	5196	5348	5368	5374	5386	5456	5464	5546	5500	5592	5600	5692
	5704	5712	5726	5807	5956	6060	6068	6083	6157	6228	6304					
?ASAVE	1235#	5460	5466	5722	5728											
?B	1280#	4413	4413	4413	4419	4459	4469	4569	4579							
?B0PNT	117#	746	754#	763	771#	780	788#	797	805#	814	822#	831	839#			
?B0R2	110#	754														
?B0R3	111#	771														
?B0R4	112#	788														
?B0R5	113#	805														
?B0R6	114#	822														
?B0R7	115#	839														
?B1PNT	126#	859	867#	880	888#	901	909#	922	930#	943	951#					
?B1R2	119#	867														
?B1R3	120#	888														
?B1R4	121#	909														
?B1R5	122#	930														
?B1R6	123#	951														
?B1R7	124#															
?BCODE	1163#	1561	1561	1561	1567	1610	1616	2390								
?BINOP	415#	1817	2387	2439	2909	3632	5179	5359	5385							
?BITSO	4217#	4411														
?BUFCN	1262#	3438	3444	3511	3517	3532	3542	3962	3968	3982	3988	4044	4054			
?BUFL	649#															
?CHARN	585#	5876														
?CHKSU	791#	3426	3426	3426	3558	3564	3571	3571	3858	3858	3858	4066	4072	4079	4079	
?CONST	104#	585	586	590	594	601	602	606	610	617	618	622	626	633	634	638
	642	649	650	654	658	671	672	676	680	686	687	691	695	701	702	706
	710	716	717	721	725	4217	4218	4222	4226							
?CURDI	912#															
?DEBNC	617#															
?DSPTI	1037#	3092	3098													
?DSPTM	808#															
?EMPHI	1136#	5308														
?EMFLO	1127#	5362														
?EPFCC	965#	2969	2975	3211	3217											
?EPFCH	1010#	2761	2777	2912	3354	3360										
?EPFCL	1001#	2734	2750	2806	2814	2819	3327	3333								
?EPFSW	974#	2792	2798	2839	2845	2894	2900	2939	2945	3249	3255	3284	3290			
?EPR0	992#	2924	2930	3268	3274	4055	4061	5060	5082							
?EPTIM	983#	2954	2960	3230	3236											
?FORM1	295#	1615	1634	1655	1688	1695	1722	1745	1780	1807	1819	1982	2000	2013	2178	2389
	2441	2445	2547	2587	2720	2735	2742	2762	2769	2793	2811	2818	2844	2862	2877	2899
	2911	2929	2944	2959	2974	3000	3009	3097	3212	3231	3250	3269	3289	3307	3328	3355
	3439	3458	3477	3496	3516	3531	3563	3584	3634	3638	3807	3814	3834	3841	3963	3987
	4002	4017	4043	4071	4263	4270	4290	4297	4322	4398	4439	4458	4550	4568	4593	4645
	4652	4765	4789	4804	4820	4842	4860	4876	4963	4987	5002	5018	5043	5061	5068	5096
	5168	5181	5197	5349	5361	5375	5387	5461	5504	5547	5581	5597	5616	5623	5693	5709
	5727	5808	5957	5971	6065	6084	6158	6229	6305							
?FORM2	319#	1638	1659	1692	1702	1986	2739	2749	2766	2776	2797	2815	2825	2866	3216	3235
	3254	3273	3311	3332	3359	3443	3462	3481	3500	3811	3821	3830	3848	3967	4267	4277
	4294	4304	4402	4649	4659	5065	5081	5465	5601	5620	5636	5713	6069			
?FORM3	339#	1755	2023	2452	2807	3541	3651	4053	4332	4449	4468	4560	4578	5520	5981	

?FORM4	356#															
?FORM5	388#	1568	1576	1611	1630	1651	1684	1718	1768	1784	1803	1978	1996	2174	2250	2331
	2466	2484	2543	2583	2716	2731	2758	2789	2887	2840	2858	2895	2925	2940	2955	2970
	3004	3065	3093	3208	3227	3246	3265	3285	3303	3324	3351	3425	3435	3454	3473	3492
	3512	3559	3580	3803	3830	3857	3959	3983	3998	4013	4067	4259	4286	4394	4412	4589
	4641	4761	4785	4800	4816	4838	4856	4872	4959	4983	4998	5014	5039	5057	5092	5164
	5193	5345	5371	5457	5543	5577	5593	5612	5657	5675	5689	5705	5723	5804	5877	5953
	6061	6080	6154	6225	6301											
?H	1298#	4262	4278	4323	4333											
?HBITH	1028#	4258	4266	4271												
?HBITL	1019#	4285	4293	4298												
?HEXBU	1324#															
?HREGA	1055#															
?HREGB	1064#															
?HREGC	1073#															
?HREGD	1082#															
?HREGE	1091#															
?HREGF	1100#															
?ITMP	774#	1687	1703	1710	1710	1717	1723	1730	1730							
?KDBBU	1208#	2332	2332	2332	2338	5615	5637	5803	5809	6153	6159					
?KEY	757#	2582	2588	2595	2595											
?KEYFL	933#															
?KEYLO	1217#	5542	5548	5658	5658	5658	5664									
?LASTK	891#	5611	5619	5624	5631	5631	5676	5676	5676							
?LDATA	740#	2810	2826	2833	2833	3633	3639	3646	3646	3652	3658	3658	4644	4660	4667	4667
	5056	5064	5069	5076	5076											
?LENGT	1333#	1388	1399#	1442	1527#	1866	1876#	1936	1946#	2146	2156#	2223	2233#	2289	2299#	2356
	2366#	2507	2517#	2640	2650#	2667	2678#	3137	3147#	3383	3398#	3606	3616#	3674	3684#	3723
	3734#	3758	3770#	3909	3920#	4095	4105#	4130	4140#	4169	4179#	4206	4233#	4348	4359#	4484
	4494#	4608	4619#	4684	4694#	4713	4723#	4906	4916#	5127	5137#	5213	5224#	5253	5263#	5298
	5300#	5402	5413#	5750	5760#	5829	5839#	5898	5908#	6022	6032#	6100	6110#	6173	6184#	6248
	6258#	6323	6335#	6391	6401#	6464	6476#	6530	6541#	6601	6612#	6675	6685#	6748	6767#	6816
	6826#	6874	6884#	6935	6945#	6996	7007#	7063	7073#	7123						
?MEMHI	1154#	3806	3822	3997	4003											
?MEMLO	1145#	3833	3849	4012	4018	4640	4648	4653								
?MINDX	156#	967	971#	971	976	980#	980	985	989#	989	994	998#	998	1003	1007#	1007
	1012	1016#	1016	1021	1025#	1025	1030	1034#	1034	1039	1043#	1043	1048	1052#	1052	1057
	1061#	1061	1066	1070#	1070	1075	1079#	1079	1084	1088#	1088	1093	1097#	1097	1102	1106#
	1106	1111	1115#	1115	1120	1124#	1124	1129	1133#	1133	1138	1142#	1142	1147	1151#	1151
	1156	1160#	1160	1165	1169#	1169	1174	1178#	1178	1183	1187#	1187	1192	1196#	1196	1201
	1205#	1205	1210	1214#	1214	1219	1223#	1223	1228	1232#	1232	1237	1241#	1241	1246	1250#
	1250	1255	1259#	1259	1264	1268#	1268	1273	1277#	1277	1282	1286#	1286	1291	1295#	1295
	1300	1304#	1304	1309	1313#	1313	1317	1321#	1321	1325	1329#	1329				
?MSAVE	158#	587	603	619	635	651	673	688	703	718	742	759	776	793	810	827
	851	872	893	914	935	967	976	985	994	1003	1012	1021	1030	1039	1048	1057
	1066	1075	1084	1093	1102	1111	1120	1129	1138	1147	1156	1165	1174	1183	1192	1201
	1210	1219	1228	1237	1246	1255	1264	1273	1282	1291	1300	1309	1317	1325	4219	
?NCOLS	601#															
?NEGL	716#	2330	5674													
?NEXTP	1199#	2251	2251	2251	2257	5870	5870	5878	5884	5952	5958	5972	5982			
?NREPT	1226#	5576	5582	5596	5602											
?NUMCO	1181#	1654	1660	2173	2179	2467	2467	2467	2473	2715	2721					
?OPTIO	1190#	1633	1639	1683	1691	1696	1783	1789	1802	1808						
?OVBUFF	1316#															
?OVSIZ	633#															
?PLUS1	686#	2465														
?PLUS3	701#	2249														

?R1	99#	4289	4305	4312	4312											
?RAM	103#	965	966	974	975	983	984	992	993	1001	1002	1010	1011	1019	1020	1028
	1029	1037	1038	1046	1047	1055	1056	1064	1065	1073	1074	1082	1083	1091	1092	1100
	1101	1109	1110	1118	1119	1127	1128	1136	1137	1145	1146	1154	1155	1163	1164	1172
	1173	1181	1182	1190	1191	1199	1200	1208	1209	1217	1218	1226	1227	1235	1236	1244
	1245	1253	1254	1262	1263	1271	1272	1280	1281	1289	1290	1298	1299			
?R80	101#	740	741	745	757	758	762	774	775	779	791	792	796	808	809	813
	825	826	830													
?R81	102#	849	850	854	858	870	871	875	879	891	892	896	900	912	913	917
	921	933	934	938	942											
?RDELN	1244#	5688	5694	5708	5714	6064	6070	6079	6085							
?RECTV	1271#	3495	3501	3579	3585											
?REGC	1289#	4397	4403	4440	4450	4551	4561	4508	4594							
?ROTCH	870#															
?ROTPH	849#	5505	5512	5512	5521	5527	5527									
?RSRVE	144#	591	595	607	611	623	627	639	643	655	659	677	681	692	696	707
	711	722	726	746	763	780	797	814	831	855	859	876	880	897	901	918
	922	939	943	4223	4227											
?SEGMA	1308#															
?SIZE	255#	1383	1437	1861	1931	2141	2218	2284	2351	2502	2635	2662	3132	3378	3601	3669
	3718	3753	3904	4090	4125	4164	4201	4343	4479	4603	4679	4708	4901	5122	5208	5248
	5293	5397	5745	5824	5893	6017	6095	6168	6243	6318	6386	6459	6525	6596	6670	6743
	6811	6869	6930	6991	7058	7118										
?SMNHI	1118#	2405	2405	2405	2491	2757	2765	2770	3457	3463	3802	3810	3815	4784	4790	4982
	4908	5182	5370	5376												
?SMALO	1109#	2730	2730	2743	2861	2867	2870	2888	3306	3312	3476	3482	3829	3837	3842	4799
	4805	4815	4821	4837	4843	4871	4877	4997	5003	5013	5019	5038	5044	5091	5097	5192
	5198	5344	5350	6224	6230	6300	6306									
?START	1339#	1383	1383	1391	1405#	1437	1437	1445	1533#	1861	1861	1869	1882#	1931	1931	1939
	1957#	2141	2141	2149	2162#	2218	2218	2226	2244#	2284	2284	2292	2310#	2351	2351	2359
	2382#	2502	2502	2510	2533#	2635	2635	2643	2656#	2662	2662	2670	2699#	3132	3132	3140
	3173#	3378	3378	3386	3414#	3601	3601	3609	3622#	3669	3669	3677	3695#	3718	3718	3726
	3745#	3753	3753	3761	3796#	3904	3904	3912	3951#	4090	4090	4098	4116#	4125	4125	4133
	4151#	4164	4164	4172	4190#	4201	4201	4209	4254#	4343	4343	4351	4385#	4479	4479	4487
	4525#	4603	4603	4611	4635#	4679	4679	4687	4700#	4708	4708	4716	4754#	4901	4901	4909
	4952#	5122	5122	5130	5158#	5208	5208	5216	5235#	5248	5248	5256	5279#	5293	5293	5301
	5334#	5397	5397	5405	5449#	5745	5745	5753	5791#	5824	5824	5832	5865#	5893	5893	5901
	5939#	6017	6017	6025	6053#	6095	6095	6103	6146#	6168	6168	6176	6220#	6243	6243	6251
	6294#	6318	6318	6326	6371#	6386	6386	6394	6437#	6459	6459	6467	6512#	6525	6525	6533
	6582#	6596	6596	6604	6653#	6670	6670	6678	6726#	6743	6743	6751	6773#	6811	6811	6819
	6832#	6869	6869	6877	6890#	6930	6930	6938	6951#	6991	6991	6999	7048#	7058	7058	7066
	7114#	7118	7118	7126												
?STRTH	1253#	1901	1907	1995	2001	2014	2024									
?TYPE	1172#	1577	1577	1577	1583	1746	1756	1769	1769	1769	1775	1820	2440	2446	2453	2542
	2548	3003	3009	3064	3070	4760	4766	4958	4964	5163	5169					
?UNARY	459#	1744	2012	2076	3530	4042	4321	4438	4457	4549	4567	5503	5970			
?VERSN	1046#															
?XPCOD	825#															
?ZERO	671#	1559	1575	1767	2483	3424	3856	5656								
BFETCH	4704	4759#														
BSRVE	1239#	5468	5730													
ASCERR	3704	3711	3715#													
B	1284#	4415	4421	4461	4529	4571										
BCODE	1167#	1563	1569	1598	1618	2392										
BITSO	4230#	4422														
BRKEND	2462	2500#														
BRKERR	3000	3000#														

BRKFIL	2433	2437#																
BRKXKT	2459#	2499																
BUFCNT	1266#	3446	3519	3534	3970	3990	4046											
BUFLN	662#	1329	3875															
BYTE11	3554	3628#																
BYTEIN	3432	3451	3470	3489	3525	3627#												
BYTE0	3995	4010	4025	4028	4039	4087	4154#											
CG0	2986	3002#																
CGONB	3019	3030#																
CGOPAT	3022	3025#																
CGOSS	3021	3034#																
CGOTRA	3023	3033#																
CGOMB	3020	3026#																
CHARCR	3393#	4119																
CHARIN	3417	3549	3699	3749#														
CHARLF	3394#	4121																
CHARNO	598#	1313	1349	1378	5640	5871	5887	5990										
CHAR0	3891	3896	3977	3980	4031	4037	4120	4122	4392#									
CHKERR	3577	3598#																
CHKSUM	803#	3428	3566	3573	3664	3665	3860	4074	4081	4155	4156							
CI0	4531#	4531	4532															
CI1	4533#	4533	4534	4536														
CI2	4537#	4585																
CI3	4539	4542#																
CI4	4541	4545#																
CIN	3749	4529#																
CKSMOK	3551	3578#																
CLEBR	1974	5870#																
CLRBFF	534#	6519	6520															
CMOINT	1836	1842	1845	1855#														
CMPMAS	3871	4673	5343#															
CMIPRET	5395#																	
CNTRL2	3395#	3418	3420	3895														
CNTTBL	3077	3080#																
CNTTRA	3083	3084	3091#															
CO1	4427#	4475																
CO2	4427	4430#																
CO3	4429	4433#																
CODEBL	199#	1390	1526	1945	2155	2232	2298	2365	2516	2649	2677	3146	3397	3615	3683	3733		
		3769	3919	4104	4139	4178	4232	4358	4493	4618	4693	4722	4915	5136	5223	5262	5307	
		5412	5759	5838	5907	6031	6109	6183	6257	6334	6400	6475	6540	6611	6684	7006	7072	
COMCBR	2407	2432#																
COMFIL	1428	1432	2426	4639#														
COMGOR	2429	2992#																
COMSER	2406	2436#																
COMSIZ	1596	1899#																
CTAB	1557	1898#																
CURDIG	928#	5478	5484	5647	5648													
DATABL	244#	1332	1875	6766	6825	6883	6944											
DATO	4029	4033#																
DAT01	4034#	4060																
DBLANK	2215	5872#	5874															
DBPNT	2036	2085#																
DBRK	1519#	1924																
DCB	2045	2106#																
DDABRK	2053	2122#																
DDAMEM	2049	2116#																

DEBNCE	630#															
DECLAR	170#	584	600	616	632	648	670	685	700	715	739	756	773	790	807	824
	848	869	890	911	932	964	973	982	991	1000	1009	1018	1027	1036	1045	1054
	1063	1072	1081	1090	1099	1108	1117	1126	1135	1144	1153	1162	1171	1180	1189	1198
	1207	1216	1225	1234	1243	1252	1261	1270	1279	1288	1297	4216				
DECSM1	5286	5291#														
DECSMA	2638	5282#														
DELAY	3105	6059#														
DELAY1	6077#	6092														
DERROR	2033	2063#														
DFILL	2040	2096#														
DGO	2039	2094#														
DGPATS	5943	5996#														
DGR	2046	2100#														
DINTRG	2051	2124#														
DLST	2041	2050#														
DMOD	2038	2092#														
DNOBRK	2055	2129#														
DONE	3419	3595#														
DPA	2050	2135#														
DPRBRK	2052	2120#														
DPRMEM	2048	2114#														
DREC	2042	2100#														
DREL	2043	2102#														
DRM	2050	2118#														
DRUN	2035	2070#														
DSB	2044	2104#														
DSGNON	2034	2070#														
DSFACC	2276	2279	2201	2320	2564	2566	5942#									
DSFHI	2268	2276#														
DSFLO	2275	2280#														
DSFMI	2273	2279#														
DSFMID	2277#	3375														
DSPTIM	1041#	3100														
DSFIMP	820#	5948	5967													
DSS	2057	2133#														
DTR	2059	2137#														
DWRK	2056	2131#														
ELSIF1	2106	2188	2202#													
ELSIF2	2204	2207	2213#													
EMHI	1140#	5390														
EMLO	1131#	5364														
ENBLNK	529#	3197	6301													
ENBRAM	528#	3197	6300													
ENDF1	3080#	3093														
ENDFIL	3072	3084#														
ENDREC	4064#															
EDFREC	3007	3901#														
EPACC	969#	2977	3219	3374	4004	5104										
EPBRK	1425	3183	6450	6456	6589#											
EPCNT	2921#	3107	3125													
EPCON1	2785	2005#														
EPCONT	2720	2783#														
EPFET	3319	3343	4052	6223#												
EPPRSS	2937	2952	2967	2982	3205	3224	3243	3262	6237	6239	6240	6298	6314	6315	6378#	
EPPCHI	1014#	2779	2914	3362	3370											
EPPCLO	1005#	2752	2821	3335												

EPPSW	978#	2800	2847	2902	2947	3257	3292						
EPR0	996#	2932	3276	4063	5084								
EPREL	3042	6445	6519#										
EPRET	3118	3122	3129#										
EPRSET	536#	1433	2994	2996	6449	6453							
EPRUN	2424	2712#											
EPRUN1	3046#	3051											
EPRUN2	3050	3062#											
EPRUN3	3049	3056#											
EPRUN4	3028	3031	3039#										
EPRUN5	3057	3115#											
EPRUN6	3081	3082	3119#										
EP55TP	532#												
EPSTE1	6447#	6448											
EPSTE2	6447	6456#											
EPSTEP	2904	3194	3198	6302	6445#								
EPSTOR	2874	2920	3340	3369	5053	6297#							
EPTIMR	987#	2962	3238										
ERROR	740	765	782	799	816	833	861	882	903	924	945		
EKROR2	2324	2349#											
EXAM0	2541#	2616											
EXAM1	2601	2618#											
EXAM2	2619	2622#											
EXAM3	2624	2627#											
EXAM4	2629	2632#											
EXAM5	2610	2613#											
EXAMIN	2410	2540#	2626	2631									
EXPNON	555#												
FDUMP1	3970	3996#											
FDUMP2	4026	4030#											
FDUMP3	4035	4038#											
FDUMP4	4064	4088#											
FDUMP5	4034	4036#											
FINDOP	1590#	1600											
GOTBL	3016	3019#											
H	1302#	4200	4325										
HBD1	4317	4319#	4319										
HBD2	4318#	4339											
HBDLAY	4257#	4433	4434	4535	4537	4538							
HBITHI	1032#	4273											
HBITLO	1023#	4300											
HDATAIN	3510#	3547											
HEXRSC	4193#	4308											
HEXBUF	1327#	3064	3075	3956	4033								
HEXNIB	4195	4198#											
HFDONE	3805	3890	3894#										
HFILE0	2413	2421	3001#	3078									
HRECIN	2416	3417#	3422	3592									
HRECO	3877	3884	3955#										
HREGA	1059#												
HREGB	1068#												
HREGC	1077#												
HREGD	1086#												
HREGE	1095#												
HREGF	1104#												
IMPLEM	1855	2305#											
INCSMR	1431	2625	3520	3073	4675	5230#							

INCH	5239#															
INCH1	5241	5246#														
INIT	1409#															
INITLP	1410#	1423														
INPAD1	2187#	2198														
INPADR	1835	2170#	2498													
INPKEY	1543	1675	1828	1846	2196	2345	2463	2580	2658#	3115	3119					
INVALS	1346#	1381	1417													
ITMP	786#	1557	1590	1595	1597	1625	1626	1646	1647	1705	1712	1715	1725	1732	1761	1830
	1832	1833	1837													
JGORE5	2408	2429#														
JMPTBL	2385	2399#														
JTOFIL	2402	2426#														
JTOGO	2401	2424#														
JTOLST	2403	2419#														
JTOMOD	2400	2410#														
JTOREC	2404	2412#														
JTOREL	2405	2416#														
KBD0UF	1212#	2334	2340	5639	5811	6161										
KBD11	5801#	5816														
KBDIN	2658	5799#														
KBDPOL	3847	3106	6150#													
KCLR8	1515#	1908														
KEY	769#	1544	1593	1740	1843	2107	2202	2205	2322	2346	2460	2590	2597	2613	2622	2627
	2659	2783	3116	3120												
KEYCLR	1505#	2323	2628													
KEYDM	1504#	1923	1926													
KEYEND	1501#	1545	1844	2206	2347	2461	2618	3117								
KEYFIL	1499#	1903														
KEYFLG	949#	5533	5671	5682												
KEYGO	1512#	1902														
KEYLOC	1221#	5550	5644	5660	5666											
KEYLST	1510#	1904														
KEYMOD	1513#	1901														
KEYNX7	1500#	2203	2623	2704	3121											
KEYPAT	1503#	1929														
KEYPM	1508#	1923	1926													
KEYREC	1506#	1905														
KEYREG	1509#	1923														
KEYREL	1502#	1906														
KEYTRA	1507#	1920														
KGORE5	1511#	1909														
KSETB	1514#	1907														
LASTKY	907#	5555	5556	5626	5633	5678										
LDATA	752#	1858	2211	2317	2327	2432	2436	2562	2565	2607	2614	2632	2628	2835	2919	3088
	3321	3344	3346	3367	3368	3526	3598	3629	3641	3648	3660	3666	3715	3868	4154	4157
	4160	4662	4669	4705	5028	5033	5071	5078	5106	5112	6297	6454				
LDBYTE	3867#	3876														
LFEBR1	4897	4899#														
LFEBRK	4780	4781	4890#													
LFEDM	4777	4797	4813	4832#												
LFINT	4779	4870#														
LFEPM	4776	4783#														
LFER0	4851	4854#														
LFEREG	4778	4836#														
LFETBL	4773	4776#														
LFETCH	2561	3867	4704#													

LFILL	4672#	4676																	
LFILL1	4674	4677#																	
LPGSEL	4832	4890	5032	5111	5162#														
LSTBR1	5113	5116#																	
LSTBR2	5115	5117#																	
LSTBRK	4978	4979	5111#																
LSTDH	4975	4995	5011	5032#															
LSTINT	4977	5090#																	
LSTORE	2459	2615	3527	4672	4957#														
LSTPM	4974	4981#																	
LSTR0	5052	5055#																	
LSTREG	4976	5037#																	
LSTTBL	4971	4974#																	
M0	551#																		
M1	552#																		
MADD	430#	1816	2386	2438	5358														
MADDC	435#	5384																	
MAIN	1434	1539#	1546	2349	2414	2417	2422	2427	2500	2620									
MAIN2	1544#	3129																	
MAINA	1594	1609#																	
MAINB	1672#	1674																	
MAINB0	1798	1830#																	
MAINB1	1831#	1847																	
MAINC1	1716#	1762																	
MAIND	1741	1801#																	
MAIND1	1742	1766#																	
MANL	440#																		
MBLOCK	165#	1307	1315	1323															
MDEC	471#	3529																	
MDJNZ	475#	4041	4320	4456	4566	5969													
MEMHI	1158#	3824	4005																
MEMLO	1149#	3851	4020	4655															
MERROR	1592	1058#																	
MINC	467#	1743	2011	2075															
MMLO1	6734#	6740																	
MMOV	398#	1558	1574	1609	1620	1649	1682	1716	1766	1782	1801	1976	1994	2172	2248	2329			
	2464	2482	2541	2581	2714	2729	2756	2787	2805	2838	2856	2893	2923	2938	2953	2968			
	3002	3063	3091	3206	3225	3244	3263	3283	3301	3322	3349	3423	3433	3452	3471	3490			
	3510	3557	3578	3601	3828	3855	3957	3981	3996	4011	4065	4257	4284	4392	4410	4587			
	4639	4759	4783	4798	4814	4836	4854	4870	4957	4981	4996	5012	5037	5055	5090	5162			
	5191	5343	5369	5455	5541	5575	5591	5610	5655	5673	5687	5703	5721	5802	5875	5951			
	6059	6078	6152	6223	6299														
MODOUT	537#	3041	6590																
MORL	445#	2908	3631	5178															
MPUSEL	553#																		
MRL	482#																		
MRLC	494#																		
MRR	486#																		
MRRC	490#	4437	4548	5502															
MXCH	455#																		
MXRL	450#																		
NCOLS	614#	5501																	
NEG1	729#	2341	5678																
NEXTPL	1203#	2253	2259	5000	5006	5960	5974												
NIBI3	3702	3708#																	
NIBIN	3627	3630	3699#																
NIBIN2	3553	3700#																	

NIBO	4159	4161	4388#																
NOBRK	1521#	1928																	
NOVALS	1381#	1416																	
NREPTS	1230#	5564	5584	5604															
NUMCON	1185#	1662	1830	2181	2469	2475	2723												
NXTLOC	5502#	5646																	
OPTAB1	1901	1903	1904	1905	1906	1922#													
OPTAB2	1907	1908	1925#																
OPTAB3	1902	1909	1927#																
OPTION	1194#	1641	1690	1791	1810														
ORGP08	128#	1400	1401	1449#	1528	1529	1873#	1947	2157	2158	2230#	2234	2300	2367	2518	2651			
	2652	2674#	2679	3148	3399	3617	3618	3681#	3685	3735	3771	3921	4106	4141	4180	4234			
	4360	4495	4620	4695	4696	4720#	4724	4917	5138	5225	5264	5309	5414	5761	5840	5909			
	6033	6111	6185	6259	6336	6402	6477	6542	6613	6686	7008	7074	7135	7152	7153				
ORGP01	129#	1952	1953	2153#	2239	2240	2296#	2305	2306	2363#	2372	2523	2684	3153	3404	3690			
	3691	3730#	3740	3741	3765#	3776	3926	4111	4112	4137#	4146	4147	4176#	4185	4186	4213#			
	4239	4365	4500	4625	4729	4922	5143	5230	5231	5260#	5269	5314	5419	5766	5845	5914			
	6038	6116	6190	6264	6341	6407	6482	6547	6610	6691	7013	7079	7136	7159	7160				
ORGP02	130#	2377	2378	2514#	2528	2529	2647#	2689	3150	3409	3410	3613#	3781	3931	4244	4370			
	4505	4630	4631	4691#	4734	4927	5148	5274	5275	5385#	5319	5424	5771	5850	5919	6043			
	6121	6195	6269	6346	6412	6487	6552	6623	6696	7018	7084	7137	7169	7170					
ORGP03	131#	1334	1335	1395#	1877	1878	1943#	6577	6578	6600#	6648	6649	6682#	6721	6722	6755#			
	6760	6769	6823#	6827	6828	6881#	6885	6886	6942#	6946	6947	7003#	7043	7044	7070#	7109			
	7110	7130#	7138	7176	7177														
ORGP04	132#	2694	2695	3144#	3163	3786	3936	4249	4250	4355#	4375	4510	4739	4932	5153	5154			
	5220#	5324	5429	5776	5855	5924	6040	6049	6107#	6126	6200	6274	6351	6417	6492	6557			
	6628	6701	7023	7089	7139	7206	7207												
ORGP05	133#	3160	3169	3390#	3791	3792	3916#	3941	4380	4381	4491#	4515	4744	4937	5329	5330			
	5409#	5434	5781	5860	5861	5905#	5929	6131	6205	6279	6356	6422	6497	6562	6633	6706			
	7028	7094	7140	7216	7217														
ORGP06	134#	3946	3947	4102#	4520	4521	4615#	4749	4750	4913#	4942	5439	5786	5787	5836#	5934			
	5935	6029#	6136	6210	6284	6361	6427	6502	6567	6638	6711	7033	7099	7141	7224	7225			
ORGP07	135#	4947	4948	5134#	5444	5445	5757#	6141	6142	6180#	6215	6216	6255#	6289	6290	6330#			
	6366	6367	6390#	6432	6433	6471#	6507	6508	6537#	6572	6643	6716	7038	7104	7142	7232			
	7233																		
OUTCLR	1624	1974#	2556																
OUTMSG	1797	1827	1975#																
OUTUTL	1542	1973#	2326	2713	2993	3124	3185												
OY00RS	3187	6783#	6784	6788	6793	6807													
OY1B1	6898	6914#																	
OY1B2	6919	6924#																	
OY1BRS	1426	3201	6451	6896#	6901	6907	6911	6914	6924										
OY2B1	6981	6984#																	
OY2BRS	2921	6957#	6958	6962	6968	6972	6984												
OY3B1	6860	6862#																	
OY3BRS	3203	6840#	6841	6845	6851	6855	6862												
OYBUF	1319#	4828	5026	6657															
OYLOAD	1427	2922	3180	3204	3202	6452	6731#												
OYSIZE	646#	1321	1426	2921	3187	3203	3201	4812	5010	6451	6657	6650	6731						
OYSW1	6661#	6667																	
OYSWAP	2905	2995	3186	6657#															
PERK	1510#	1924																	
PDIGIT	517#	5480																	
PERROR	1859	2212	2310#	2633	3009	3599	3716	6455											
PGSIZE	7135#	7136#	7137#	7138#	7139#	7140#	7141#	7142#											
PINPUT	520#	5481																	
PLUS1	699#	2476																	

PLUS3	714#	2260																	
PRNT1	2009	2030#																	
PRNT2	1994#	2029																	
PSEGH1	518#	1414	5476	5491	5818														
PSEGLO	519#	1413	5477	5489	5819														
RDELAY	1248#	5696	5716	6072	6007														
RECDON	3524	3549#																	
RECTYP	1275#	3503	3587																
REGC	1293#	4405	4442	4553	4596														
REORG	191#	1335	1401	1529	1878	1948	1953	2158	2235	2240	2301	2306	2368	2373	2378	2519			
	2524	2529	2652	2680	2685	2690	2695	3149	3154	3159	3164	3169	3400	3405	3410	3618			
	3686	3691	3736	3741	3772	3777	3782	3787	3792	3922	3927	3932	3937	3942	3947	4107			
	4112	4142	4147	4181	4186	4235	4240	4245	4250	4361	4366	4371	4376	4381	4496	4501			
	4506	4511	4516	4521	4621	4626	4631	4696	4725	4730	4735	4740	4745	4750	4918	4923			
	4928	4933	4938	4943	4948	5139	5144	5149	5154	5226	5231	5265	5270	5275	5310	5315			
	5320	5325	5330	5415	5420	5425	5430	5435	5440	5445	5762	5767	5772	5777	5782	5787			
	5841	5846	5851	5856	5861	5910	5915	5920	5925	5930	5935	6034	6039	6044	6049	6112			
	6117	6122	6127	6132	6137	6142	6186	6191	6196	6201	6206	6211	6216	6260	6265	6270			
	6275	6280	6285	6290	6337	6342	6347	6352	6357	6362	6367	6403	6408	6413	6418	6423			
	6428	6433	6478	6483	6488	6493	6498	6503	6508	6543	6548	6553	6558	6563	6568	6573			
	6578	6614	6619	6624	6629	6634	6639	6644	6649	6687	6692	6697	6702	6707	6712	6717			
	6722	6769	6828	6886	6947	7009	7014	7019	7024	7029	7034	7039	7044	7075	7080	7085			
	7090	7095	7100	7105	7110														
RERROR	2317#	2348																	
RINT	1520#	1923																	
ROTCN1	886#	5501	5646																
ROTPAT	065#	5482	5507	5514	5529														
RSOURC	276#	7133																	
SCAN3	5557	5575#																	
SCAN5	5532	5566	5589	5609	5644#														
SCAN8	5672	5682#																	
SEGMAP	1311#	2213	5486	5870	5965														
SING	1523#	1928																	
SIZE	1385#	1388	1439#	1442	1863#	1866	1933#	1936	2143#	2146	2220#	2223	2286#	2289	2353#	2356			
	2504#	2507	2637#	2640	2664#	2667	3134#	3137	3380#	3383	3603#	3606	3671#	3674	3720#	3723			
	3755#	3758	3906#	3909	4092#	4095	4127#	4130	4166#	4169	4203#	4206	4345#	4348	4481#	4484			
	4605#	4608	4681#	4684	4710#	4713	4903#	4906	5124#	5127	5210#	5213	5250#	5253	5295#	5298			
	5399#	5402	5747#	5750	5826#	5829	5895#	5898	6019#	6022	6097#	6100	6170#	6173	6245#	6248			
	6320#	6323	6388#	6391	6461#	6464	6527#	6530	6590#	6601	6672#	6675	6745#	6748	6813#	6816			
	6871#	6874	6932#	6935	6993#	6996	7060#	7063	7120#	7123									
SIZECH	270#	1382	1436	1860	1930	2140	2217	2283	2350	2501	2634	2661	3131	3377	3600	3668			
	3717	3752	3903	4009	4124	4163	4200	4342	4478	4602	4678	4707	4900	5121	5207	5247			
	5292	5396	5744	5823	5892	6016	6094	6167	6242	6317	6385	6458	6524	6595	6669	6742			
	6810	6868	6929	6990	7057	7117													
SMAPHI	1122#	2487	2493	2772	3465	3817	4792	4990	5184	5378									
SMALO	1113#	1671	1831	2480	2557	2745	2869	2880	3314	3341	3484	3844	4807	4823	4845	4879			
	5005	5021	5046	5099	5200	5238	5282	5352	6232	6308									
STRCOM	1623	2037#																	
STRGOC	1927	2054#																	
STRMEM	1922	1925	2047#	2555															
STRTMP	1257#	1989	2003	2016															
STRUTL	1973	2032#																	
STSRVE	3056	3062	3183#																
TCRLFO	3886	3094	3975	4119#															
TIINT	5454#	5742																	
TIRET1	5647	5701	5721#																
TOFPOL	3046	5742#	5801	6077															

TTYOUT	539#	4428	4430												
TYPE	1176#	1429	1579	1585	1748	1771	1777	1822	2448	2550	3011	3072	4768	4966	5171
UPDH01	2265#	2558	3371												
UPDADR	2195	2248#													
VERNO	1050#														
WRK	1522#	1928													
WDISP	2010	2030	2269	2274	2560	3373	5948#								
WDISP1	5988	5991#													
XPCODE	837#	1410	1539	2318	5799	5949	6150								
XPTST	1411	1540	2319	5800	5950	6151	7050#								
ZERO	684#	1570	1586	1778	2494	3428	3860	5667							

CROSS REFERENCE COMPLETE

APPENDIX C COMMAND SUMMARY

The following is a summary of the commands implemented by the HSE-49 emulator monitor. Within each command group, tokens in each column indicate options the user has when invoking those commands.

Tokens in square brackets indicate dedicated keys on the keyboard (some keys having shared functions); angle brackets enclose hex digit strings used to specify an address or data parameter. Parameters in parentheses are optional, with the effects explained above. The notation used is as follows:

<SMA> — Starting Memory Address for block command,
<EMA> — Ending Memory Address for block command,
<LOC> — LOCAtion for individual accesses,
<DATA> — DATA byte.

Asterisks (*) indicate the default condition for each command; thus that token is optional and serves to regularize the command syntax.

Program/data entry and verification commands:

```
[EXAM]  [PROG MEM]* <LOC> [.] [NEXT]
        [DATA MEM]          [PREV]
        [REGISTER]          [.]
        [HWRE REG]
        [PROG BRK]
        [DATA BRK]
```

Program/data initialization commands:

```
[FILL]  [PROG MEM]* <SMA> [.] <EMA> [.] <DATA> [.]
        [DATA MEM]
        [REGISTER]
        [HWRE REG]
        [PROG BRK]
        [DATA BRK]
```

Intellec® development system or TTY interface commands (for transferring HEX format files):

```
[UPLOAD] [PROG MEM]* <SMA> [.] <EMA> [.]
        [DATA MEM]
        [REGISTER]
        [HWRE REG]
        [PROG BRK]
        [DATA BRK]

[DNLOAD] [PROG MEM]* [.]
        [DATA MEM]
        [REGISTER]
        [HWRE REG]
        [PROG BRK]
        [DATA BRK]
```

Formatted data dump to TTY or CRT:

```
[LIST]  [PROG MEM]* <SMA> [.] <EMA> [.]
        [DATA MEM]
        [REGISTER]
        [HWRE REG]
        [PROG BRK]
        [DATA BRK]
```

Program execution commands:

```
[GO]     [NO BREAK]* <SMA> [.]
        [W/ BREAK]          [.]
        [SING STP]
        [AUTO BRK]
        [AUTO STP]

[GO/RST] [NO BREAK]* [.]
        [W/ BREAK]
        [SING STP]
        [AUTO BRK]
        [AUTO STP]
```

Breakpoint setting and clearing:

```
[SET BRK] [PROG MEM]* <LOC> ([.] <LOC> ... ) [.]
        [DATA MEM]

[CLR BRK] [PROG MEM]* <LOC> ([.] <LOC> ... ) [.]
        [DATA MEM]
```

APPENDIX D ERROR MESSAGES

The following error message codes are used by the monitor software to report an operator or hardware error. Errors may be cleared by pressing [CLR/PREV] or [END/]. The format used for reporting errors is "Error — .n" where "n" is a hex digit.

Operator Errors

1. Illegal command initiator.
2. Illegal command modifier or parameter digit.
3. Illegal terminator for Examine command.
4. Illegal attempt to clear Error mode.
- 5–9. Not used.

Hardware Errors

- A. ASCII error — non-hex digit encountered in data field of hex format record.
- B. Breakpoint error. Break logic activated though breakpoints not enabled.
- C. Hex format record checksum error. Note — the checksum will not be verified if the first character of the checksum field is a question mark (" ") rather than a hexadecimal digit. This allows object files to be patched using the ISIS text editor without the necessity of manually recomputing the checksum value.
- D. Not used.
- E. Execution processor failed to respond to a command or parameter passed to it by the master processor. EP automatically reset. EP internal status may be lost. Program memory not affected.
- F. Not used.



U.S. AND CANADIAN SALES OFFICES

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

ALABAMA

Intel Corp.
3322 S. Parkway, Ste. 71
Holiday Office Center
Huntsville 35802
Tel: (205) 883-2430
†Pen-Tech Associates, Inc.
Holiday Office Center
3322 Memorial Pkwy., S.W.
Huntsville 35801
Tel: (205) 881-9298

ARIZONA

Intel Corp.
8650 N. 35th Avenue, Suite 101
Phoenix 85021
Tel: (602) 242-7205

†BFA

4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA

Intel Corp.
7670 Opportunity Rd.
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*

1651 East 4th Street
Suite 105
Santa Ana 92701
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*

15335 Morrison
Suite 345
Sherman Oaks 91403
(213) 986-9510
TWX: 910-495-2045

Intel Corp.*

3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
TWX: 910-338-0255
Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I

2576 Shattuck Ave.
Suite 4B
Berkeley 94704
Tel: (415) 843-7625

Mac-I

P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I

P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I

110 Sutter Street
Suite 715
San Francisco 94104
Tel: (415) 982-3673

Mac-I

20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO

Intel Corp.*
6000 East Evans Ave.
Bldg. 1, Suite 260
Denver 80222
Tel: (303) 758-8086
TWX: 910-931-2289

†Westek Data Products, Inc.
25921 Fern Gulch

P.O. Box 1355
Evergreen 80439
Tel: (303) 674-5255

Westek Data Products, Inc.
1322 Arapahoe
Boulder 80302
Tel: (303) 449-2620

CONNECTICUT

Intel Corp.
Peacock Alley
1 Padanaram Road, Suite 146
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

FLORIDA

Intel Corp.
1001 N.W. 62nd Street, Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

FLORIDA (cont.)

Intel Corp.
5151 Adanson Street, Suite 203
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

†Pen-Tech Associates, Inc.
201 S.E. 15th Terrace, Suite K
Deerfield Beach 33441
Tel: (305) 421-4989

†Pen-Tech Associates, Inc.
111 So. Maitland Ave., Suite 202
P.O. Box 1475
Maitland 32751
Tel: (305) 645-3444

GEORGIA

Pen Tech Associates, Inc.
Cherokee Center, Suite 21
627 Cherokee Street
Marietta 30060
Tel: (404) 424-1931

ILLINOIS

Intel Corp.*
900 Jorie Boulevard
Suite 220
Oakbrook 60521
Tel: (312) 325-9510
TWX: 910-651-5881

First Rep Company
9400-9420 W. Foster Avenue
Chicago 60656
Tel: (312) 992-0830
TWX: 910-227-4927

Technical Representatives
1502 North Lee Street
Bloomington 61701
Tel: (309) 829-8080

IOWA

Technical Representatives, Inc.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52405
Tel: (319) 393-5510

KANSAS

Intel Corp.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

Technical Representatives, Inc.
8245 Nieman Road, Suite #100
Lenexa 66214
Tel: (913) 888-0212, 3, & 4
TWX: 910-749-6412

KENTUCKY

Lowry & Associates Inc.
P.O. Box 5127
Lexington 40555
Tel: (606) 273-3771

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944
Glen White Associates
57 W. Timonium Road, Suite 307
Timonium 21093
Tel: (301) 252-6360

†Mesa Inc.
11900 Parklawn Drive
Rockville 20852
Tel: Wash. (301) 881-8430
Balt. (301) 792-0021

MASSACHUSETTS

Intel Corp.*
27 Industrial Ave.
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333
EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740

MICHIGAN

Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 910-420-1212
TELEX: 2 31143
†Lowry & Associates, Inc.
135 W. North Street
Suite 4
Brighton 48116
Tel: (313) 227-7067

MINNESOTA

Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

†Dytek North
1821 University Ave.
Room 163N
St. Paul 55104
Tel: (612) 645-5816

MISSOURI

Technical Representatives, Inc.
320 Brookes Drive, Suite 104
Hazelwood 63042
Tel: (314) 731-5200
TWX: 910-762-0618

NEW JERSEY

Intel Corp.*
1 Metroplaza Office Bldg.
505 Thornall St.
Edison 08817
Tel: (201) 494-5040
TWX: 710-480-6238

NEW MEXICO

BFA Corporation
P.O. Box 1237
Las Cruces 88001
Tel: (505) 523-0601
TWX: 910-983-0543
BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK

Intel Corp.*
350 Vanderbilt Motor Pkwy.
Suite 402
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.

80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.

2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 328-7340
TWX: 510-253-3841

†Measurement Technology, Inc.

159 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared

4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared

2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

†Pen-Tech Associates, Inc.
1202 Eastchester Dr.
Highpoint 27260
Tel: (919) 883-9125

Glen White Associates
4009 Barrett Dr.
Raleigh 27609
Tel: (919) 787-7016

OHIO

Intel Corp.*
8312 North Main Street
Dayton 45415
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*

Chagrin-Brainard Bldg. #210
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736

Lowry & Associates Inc.
1440 Snow Road
Suite 216
Cleveland 44134
Tel: (216) 398-0506

†Lowry & Associates, Inc.
1524 Marsetta Drive
Dayton 45432
Tel: (513) 429-9040

†Lowry & Associates, Inc.
1050 Freeway Dr., N.
Suite 209
Columbus 43229
Tel: (614) 436-2051

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086

PENNSYLVANIA

Intel Corp.*
275 Commerce Dr.
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-661-2077

†Lowry & Associates, Inc.

Seven Parkway Center
Suite 455
Pittsburgh 15520
Tel: (412) 922-5110

†Q.E.D. Electronics

300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS

Intel Corp.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5487

Intel Corp.*

6420 Richmond Ave.
Houston 77057
Tel: (713) 784-3400
Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421
Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

VIRGINIA

Glen White Associates
Route 2, Box 193
Charlottesville 22901
Tel: (804) 295-7686

Glen White Associates
P.O. Box 10186
Lynchburg 24506
Tel: (804) 384-6920

Glen White Associates
Rt. #1, Box 322
Colonial Beach 22442
Tel: (804) 224-7764

WASHINGTON

Intel Corp.
Suite 114 Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086

WISCONSIN

Intel Corp.
4369 S. Howell Ave.
Milwaukee 53207
Tel: (414) 747-0789

CANADA

Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
6205 Airport Rd.
Bldg. B, Suite 205
Mississauga, Ontario
L4V 1E3
Tel: (416) 671-0611
TELEX: 06983574

Multitek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multitek, Inc.
Toronto
Tel: (416) 245-6422
Multitek, Inc.
Montreal
Tel: (514) 481-1350

*Field application location

†These representatives do not offer Intel Components, only boards and systems.

98-01014-01



3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

ALABAMA

†Hamilton/Avnet Electronics
4692 Commercial Drive N.W.
Huntsville 35805
Tel: (205) 837-7210
Pioneer
1207 Putman Drive NW
Huntsville 35805
Tel: (205) 837-9033
TWX: 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
2615 South 21st Street
Phoenix 85034
Tel: (602) 275-7851
†Liberty/Arizona
8155 N. 24th Avenue
Phoenix 85021
Tel: (602) 249-2232
TWX: 910-951-4282

CALIFORNIA

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6111
TWX: 910-595-1928
Hamilton/Avnet
1175 Bordeaux Dr.
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332
†Hamilton/Avnet Electronics
8917 Complex Drive
San Diego 92123
Tel: (714) 279-2421
TWX: 910-335-1216
Hamilton/Avnet
10912 W. Washington Blvd.
Culver City 90230
Tel: (213) 558-2809 (2665)
TWX: 910-340-6364 or 7073
†Hamilton Electro Sales
10912 W. Washington Boulevard
Culver City 90230
Tel: (213) 558-2121
†Liberty Electronics
124 Maryland Street
El Segundo 90245
Tel: (213) 322-3826
TWX: 910-348-7140 or 7111
†Liberty/San Diego
9525 Chesapeake Dr.
San Diego 92123
Tel: (714) 565-9171
TWX: 910-335-1590
†Wyle/Elmar
3000 Bowers Avenue
Santa Clara 95052
Tel: (408) 727-2500
TWX: 910-338-0451 or 0296
Hamilton/Avnet Electronics
17312 Eastman Street
Irvine 92714
Tel: (714) 979-6864

COLORADO

†Wyle/Elmar/Denver
6777 E. 50th Avenue
Commerce City 80022
Tel: (303) 287-9611
TWX: 910-931-0510
†Hamilton/Avnet Electronics
5921 No. Broadway
Denver 80216
Tel: (303) 534-1212
TWX: 910-931-0510

CONNECTICUT

†Cramer/Connecticut
P.O. Box 5003
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162
†Hamilton/Avnet Electronics
643 Danbury Road
Georgetown 06829
Tel: (203) 762-0361
†Harvey Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373

FLORIDA

Arrow Electronics
1001 N.W. 62nd Street
Suite 108
Ft. Lauderdale 33309
Tel: (305) 776-7790
Arrow Electronics
115 Palm Bay Road, NW
Suite 10, Bldg. 200
Palm Bay 32905
Tel: (305) 725-1480
TWX: 510-959-6337
†Hamilton/Avnet Electronics
6800 Northwest 20th Ave.
Ft. Lauderdale 33309
Tel: (305) 971-2900
TWX: 510-955-3097
†Pioneer
6220 S. Orange Blossom Trail
Suite 412
Orlando 32809
Tel: (305) 859-3600
TWX: 810-850-0177
Hamilton/Avnet
3197 Tech. Drive N.
St. Petersburg 33702
Tel: (813) 576-3930
TWX: 810-863-0374

GEORGIA

Arrow Electronics
3406 Oak Cliff Road
Doraville 30340
Tel: (404) 455-4054
TWX: 810-757-4213
†Hamilton/Avnet Electronics
6700 I-85 Access Road, #11
Suite 1E
Norcross 30071
Tel: (404) 448-0800

ILLINOIS

Arrow Electronics
492 Lunt Avenue
P.O. Box 94248
Schaumburg 60193
Tel: (312) 893-9420
TWX: 910-222-1807
†Hamilton/Avnet Electronics
3901 No. 25th Ave.
Schiller Park 60176
Tel: (312) 678-6310
TWX: 910-227-0060
Pioneer/Chicago
1511 Carmen Drive
Elk Grove 60006
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

KANSAS

†Hamilton/Avnet Electronics
9219 Quivira Road
Overland Park 66215
Tel: (913) 888-8900

MARYLAND

†Hamilton Avnet
P.O. Box 647,
BWI Airport
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5684
TWX: 710-862-1861
†Pioneer/Washington
9100 Gaither Road
Gaithersburg 20760
Tel: (301) 948-0710
TWX: 710-828-0545

MASSACHUSETTS

†Cramer Electronics Inc.
85 Wells Avenue
Newton 02159
Tel: (617) 969-7700

MASSACHUSETTS (continued)

†Hamilton/Avnet Electronics
50 Tower Office Park
Woburn 01801
Tel: (617) 935-9700
Harvey/Boston
44 Hartwell Ave.
Lexington 02173
Tel: (617) 861-9200

MICHIGAN

†Arrow Electronics
3921 Varsity Road
Ann Arbor 48140
Tel: (313) 971-8220
TWX: 810-223-6020
†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
TWX: 810-242-3271
†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

MINNESOTA

†Industrial Components
5229 Edina Industrial Blvd.
Minneapolis 55435
Tel: (612) 831-2666
TWX: 910-576-3153
Arrow Electronics
5251 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-2726
†Hamilton/Avnet Electronics
7449 Cahill Road
Edina 55435
Tel: (612) 941-3801
TWX: 910-576-2720

MISSOURI

†Hamilton/Avnet Electronics
396 Brookes Drive
Hazelwood 63042
Tel: (314) 731-1144
TWX: 910-762-0606

NEW JERSEY

Arrow/Philadelphia
Pleasant Valley
Moorestown 08057
Tel: (215) 928-1800
TWX: 710-897-0829
Arrow Electronics
285 Midland Avenue
Saddlebrook 07662
Tel: (201) 797-5800
TWX: 710-988-2206
Hamilton/Avnet
10 Industrial
Fairfield 07006
Tel: (201) 575-3390
TWX: 710-734-4338
†Harvey Electronics
45 Route 46
Pinebrook 07058
Tel: (201) 227-1262
TWX: 710-734-4382
†Hamilton/Avnet Electronics
1 Keystone Ave.
Cherry Hill 08003
Tel: (609) 424-0100
TWX: 710-897-1405

NEW MEXICO

†Alliance Electronics Inc.
11721 Central Ave.
Albuquerque 87123
Tel: (505) 292-3360
TWX: 910-989-1151
†Hamilton/Avnet Electronics
2524 Baylor Drive, S.E.
Albuquerque 87119
Tel: (505) 765-1500



3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

NEW YORK

Harvey Electronics
P.O. Box 1208
Binghamton 13902
Tel: (607) 748-8211
TWX: 510-252-0893

Arrow Electronics
900 Broad Hollow Road
Farmingdale 11735
Tel: (516) 694-6800
TWX: 510-224-6494

†Cramer/Rochester
3000 South Winton Road
Rochester 14623
Tel: (716) 275-0300
TWX: 910-338-0026

†Hamilton/Avnet Electronics
167 Clay Road
Rochester 14623
Tel: (716) 442-7820
TWX: 910-340-6364

†Cramer/Syracuse
7705 Maitlage Drive
Liverpool 13088
Tel: (315) 652-1000
TWX: 710-545-0230

Arrow Electronics
399 Conklin Street
Farmingdale 11735
Tel: (516) 694-6800
TWX: 510-224-6494

†Hamilton/Avnet Electronics
16 Corporate Circle
E. Syracuse 13057
Tel: (315) 437-2641

†Hamilton/Avnet Electronics
70 State Street
Westbury, L.I. 11590
Tel: (516) 545-6000
TWX: 510-252-0893

†Harvey Electronics
60 Crossways Park West
Woodbury 11797
Tel: (516) 921-8700
TWX: 510-221-2184
Harvey/Rochester
10 Riverton Way
West Henrietta 14586
Tel: (716) 334-5920

NORTH CAROLINA

Pioneer/Carolina
106 Industrial Ave.
Greensboro 27406
Tel: (919) 273-4441
TWX: 510-925-1114

†Hamilton/Avnet Electronics
2803 Industrial Drive
Raleigh 27609
Tel: (919) 829-8030

Arrow Electronics
P.O. Box 989
Kernersville 27284
Tel: (919) 996-2039
TWX: 510-922-4765

OHIO

Arrow Electronics
3100 Plainfield Road
Kettering 45432
Tel: (513) 253-9176
TWX: 810-459-1611

Arrow Electronics
6238 Cochran Rd.
Solon 44139
Tel: (216) 248-3990

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 433-0610
TWX: 910-340-2531

†Pioneer/Dayton
1900 Troy Street
Dayton 45404
Tel: (513) 236-9900
TWX: 810-459-1622

Arrow Electronics
10 Knollcrest Dr.
Reading 44139
Tel: (513) 761-5432
TWX: 810-461-2670

OHIO (continued)

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
TWX: 810-422-2210

†Hamilton/Avnet Electronics
4588 Emory Industrial Parkway
Warrensville Heights 44128
Tel: (216) 831-3500

OKLAHOMA

†Components Specialties, Inc.
7920 E. 40th Street
Tulsa 74145
Tel: (918) 664-2820
TWX: 910-845-2215

OREGON

†Almac/Stroum Electronics
8022 S.W. Nimbos, Bldg. 7
Beaverton 97005
Tel: (503) 641-9070

PENNSYLVANIA

Arrow/Pittsburgh
4297 Greensburg Pike
Suite 3114
Pittsburgh 15221
Tel: (412) 351-4000

Pioneer/Pittsburgh
560 Alpha Drive
Pittsburgh 15238
Tel: (412) 782-2300
TWX: 710-795-3122

Pioneer/Delaware Valley
141 Gibraltar Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TENNESSEE

Arrow/Knoxville
P.O. Box 129
W. Andrew Johnson Hwy.
Talbott 37677
Tel: (615) 587-2137

TEXAS

Component Specialties Inc.
8222 Jamestown Drive
Suite 115
Austin 78758
Tel: (512) 837-8922
TWX: 910-874-1320

Hamilton/Avnet
10508A Boyer Blvd.
Austin 78757
Tel: (512) 837-8911

†Hamilton/Avnet Electronics
4445 Sigma Road
Dallas 75240
Tel: (214) 661-8661
TWX: 910-860-5371

†Hamilton/Avnet Electronics
3939 Ann Arbor
Houston 77063
Tel: (713) 780-1771

†Component Specialties, Inc.
10807 Shady Trail, Suite 101
Dallas 75220
Tel: (214) 357-6511
TWX: 910-861-4999

†Component Specialties, Inc.
8585 Commerce Park Drive, Suite 590
Houston 77036
Tel: (713) 771-7237
TWX: 910-881-2422

Arrow Electronics
13715 Gamma Road
Dallas 75234
Tel: (214) 386-7500
TWX: 910-861-5495

UTAH

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City, 84119
Tel: (801) 972-2800

WASHINGTON

†Hamilton/Avnet Electronics
14212 N.E. 21st
Bellevue 98005
Tel: (206) 746-8750

WASHINGTON (continued)

†Almac/Stroum Electronics
5811 Sixth Ave. South
Seattle 98108
Tel: (206) 763-2300
TWX: 910-444-2067

†Liberty Electronics
1750 132nd Avenue NE
Bellevue 98005
Tel: (206) 453-8300
TWX: 910-443-2526

WISCONSIN

Arrow Electronics
434 W. Rausson Avenue
Oak Creek 53154
Tel: (414) 764-6600
TWX: 910-338-0026

†Hamilton/Avnet
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX: 910-262-1182

CANADA

ALBERTA

†L. A. Varah Ltd.
4742 14th Street N.E.
Calgary T2E 6L7
Tel: (403) 230-1235
TWX: 018-258-97

BRITISH COLUMBIA

†L.A. Varah Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068

Zenronics
550 Cambie St.
Vancouver V6B 2N7
Tel: (604) 688-2533
TWX: 04-5077-89

MANITOBA

L. A. Varah
1-1832 King Edward Street
Winnipeg R2P 0N1
Tel: (204) 633-6190
TWX: 07-55-365

ONTARIO

†L.A. Varah, Ltd.
505 Kenora Avenue
Hamilton L8E-3P2
Tel: (416) 561-9311
TWX: 061-8349

†Hamilton/Avnet Electronics
3688 Nashua Drive, Units G & H
Mississauga L4V 1M5
Tel: (416) 677-7432
TWX: 610-492-8860

†Hamilton/Avnet Electronics
1735 Courtwood Cresc.
Ottawa K2C 3J2
Tel: (613) 226-1700

†Zenronics
141 Catherine Street
Ottawa, Ontario K2P 1C3
Tel: (613) 238-6411
TWX: 053-3636

†Zenronics
1355 Meyerside Drive
Mississauga, Ontario L5T 1C9
Tel: (416) 676-9000
Telex: 06-983-657

QUEBEC

†Hamilton/Avnet Electronics
2670 Paulus Street
St. Laurent H4S 1G2
Tel: (514) 331-6443
TWX: 610-421-3731

Zenronics
5010 Pare Street
Montreal H4P 1P3
Tel: (514) 735-5361
TWX: 05-827-535



INTERNATIONAL SALES AND MARKETING OFFICES

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-0800
TWX: 910 338-0026
TELEX: 34-6372

EUROPEAN MARKETING OFFICES

BELGIUM
Intel International
Rue du Moulin à Papier 51
Boite 1
B-1160 Brussels
Tel: (02) 660 30 10
TELEX: 24814

DENMARK
Intel Denmark A/S*
Lyngbyvej 32 2nd Floor
DK-2100 Copenhagen East
Tel: (01) 18 20 00
TELEX: 19567

FINLAND
Intel Scandinavia
Sennierikujä 3
SF-00400 Helsinki 40
Tel: (0) 558531
TELEX: 123 332

FRANCE
Intel Corporation, S.A.R.L.*
5 Place de la Balance
Silo 223
94508 Rungis Cedex
Tel: (01) 687 22 21
TELEX: 270475

GERMANY
Intel Semiconductor GmbH*
Seidlstrasse 27
8000 München 2
Tel: (089) 53 691
TELEX: 523 177
Intel Semiconductor GmbH
Abraham Lincoln Strasse 30
6200 Wiesbaden 1
Tel: (06121) 74855
TELEX: 04186183
Intel Semiconductor GmbH
Wernerstrasse 67
P.O. Box 1460
7012 Fellbach
Tel: (0711) 580082
TELEX: 7254826
Intel Semiconductor GmbH
Hindenburgstrasse 78/29
3000 Hannover
Tel: (0511) 852051
TELEX: 923625

ISRAEL
Intel Semiconductor Ltd.*
P.O. Box 2404
Haifa
Tel: 972/452 4261
TELEX: 92246511

ITALY
Intel Corporation Italia, s.p.a.
Corso Sempione 39
I-20145 Milano
Tel: 2/34.93287
TELEX: 311271

NETHERLANDS
Intel Semiconductor Nederland B.V.
Cometongebouw
Westblaak 106
3012 Km Rotterdam
Tel: (10) 149122
TELEX: 22283

NORWAY
Intel Scandinavia A/S
P.O. Box 158
N-2040
Kløfta, Norway
Tel: 47 2/981068
TELEX: 18018

SWEDEN
Intel Sweden AB*
Box 20092
Enighetsvagen, 5
S-16120 Bromma
Tel: (08) 98 53 90
TELEX: 12261

SWITZERLAND
Intel Semiconductor AG
Forchstrasse 95
CH 8032 Zurich
Tel: 1-55 45 02
TELEX: 557 89 ich ch

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.*
Broadfield House
4 Between Towns Road
Cowley, Oxford OX4 3NB
Tel: (0865) 77 14 31
TELEX: 837203
Intel Corporation (U.K.) Ltd.
5 Hospital Street
Nantwich, Cheshire CW5 5RE
Tel: (0270) 62 65 60
TELEX: 36620

ORIENT MARKETING OFFICE

JAPAN
Intel Japan K.K.*
Flower Hill-Shinmachi East Bldg.
1-23-9, Shinmachi, Setagaya-ku
Tokyo 154
Tel: (03) 426-9261
TELEX: 781-28426

HONG KONG
Intel Trading Corporation
99-105 Oes Voeux Rd., Central
18F, Unit B
Hong Kong
Tel: 852-5-450-847
TELEX: 63869

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA
Micro Sistemas S.A.
9 De Julio 561
Cordoba
Tel: 54-51-32-880
TELEX: 51837 BICCO

AUSTRALIA
A.J.F. Systems & Components Pty. Ltd.
Corporate Headquarters
44 Prospect Rd.
Prospect
South Australia 5082
Tel: 268-1244
TELEX: 82635
Wairburton Franki
Corporate Headquarters
372 Eastern Valley Way
Chatswood, New South Wales 2067
Tel: 407-3261
TELEX: AA 21299

AUSTRIA
Bacher Elektronische Geräte GmbH
Rotenmühlgasse 26
A 1120 Vienna
Tel: (0222) 83 63 96
TELEX: (01) 1532
Rekirsch Elektronik Geräte GmbH
Lichteneinsstrasse 97
A1090 Vienna
Tel: (222) 347646
TELEX: 74759

BELGIUM
Intelco Belgium S.A.
Avenue Val Duchesse, 3
B-1180 Brussels
Tel: (02) 660 00 12
TELEX: 25441

BRAZIL
Icotron S.A.
05110-Av. Mulinga 3650
6 Andar
Piniluba-Sao Paulo
Tel: 261-0211
TELEX: (011) 222 1CO BR

COLOMBIA
International Computer Machines
Adpo. Aero 19403
Bogota 1
Tel: 232-6635
TELEX: 43439

DENMARK
Lyngso Komponent A/S
Ostmarken 4
DK-2860 Soborg
Tel: (01) 67 00 77
TELEX: 22990
Scandinavian Semiconductor
Supply A/S
Nannasgade 18
DK-2200 Copenhagen N
Tel: (01) 83 50 90
TELEX: 19037

FINLAND
Oy Fintron AB
Meikonkatu 24 A
SF-00210
Helsinki 21
Tel: 0-692 6022
TELEX: 124 224 Ftron SF

FRANCE
Celdis S.A.
53, Rue Charles Frerol
94250 Gentilly
Tel: (1) 581 00 20
TELEX: 200 485 F
Metrologie*
La Tour d'Asnières
4, Avenue Laurent Cely
92606-Asnières
Tel: 791 44 44
TELEX: 611 448 F
Metrologie
36 Rue Tronchet
69006 Lyon
Tel: (79) 89 35-65
Tekelec Airtronic*
Cite des Bruyeres
Rue Carle Vernet
92310 Sevres
Tel: (1) 027 75 35
TELEX: 204552

Tekeltec Airtronic
75 Rue Bataille
69008 Lyon
Tel: (78) 74-37-40
TELEX: 370 481 F
Tekelec Airtronic
Parc Industriel Bertol
Voie Romaine
33600 Peyssac
Tel: (56) 45 32-27
TELEX: 570 264 F
Tekelec Airtronic
Allée des Lilas
13100 Aix en Provence
Tel: (42) 27-66-45
TELEX: 420 568 F
Tekelec Airtronic
8 Rue de l'Université
67000 Strasbourg
Tel: (88) 35-69-22
TELEX: 880 765 F
Tekelec Airtronic
281 Route d'Espagne
31076 Toulouse
Tel: (61) 40-24-90/40-38-77

GERMANY
Electronic 2000 Vertriebs GmbH
Neumarkter Strasse 75
D-8000 München 80
Tel: (089) 434061
TELEX: 522561
Jermyn GmbH
Postfach 1180
D-6277 Camberg
Tel: (06434) 231
TELEX: 484426
Kontron Elektronik GmbH
Breslauerstrasse 2
8057 Eching 1
D-8500 München
Tel: (89) 319 011
TELEX: 522122
Alfred Neye Enatechnik GmbH
Schillerstrasse 14
D-2085 Quickborn-Hamburg
Tel: (04106) 6121
TELEX: 02-13690

HONG KONG
Schmidt & Co.
28/F Wing on Center
Connaught Road
Hong Kong
Tel: 5-455-644
TELEX: 74766 Schmc Hx

INDIA
Micro Devices
104/109C, Nirmal Industrial Estate
Sion (E)
Bombay 400022, India
Tel: 485-170
TELEX: 011-5947 MDEV IN

ISRAEL
Electronics Ltd.*
11 Rozans Street
P.O. Box 39300
Tel-Aviv 61390
Tel: 475151
TELEX: 33638

ITALY
Eledra 3S S.P.A.*
Viale Elvezio, 18
20154 Milan
Tel: 316 54 41
TELEX: 39332
Eledra 3S S.P.A.*
Via Paolo Gaidano, 141 D
10137 Torino
Tel: (011) 30 97 097 - 30 97 114
TELEX: 210632
Eledra 3S S.P.A.
Via Zaccaroni Alvisi 6
40100 Bologna
Tel: (051) 307781
Eledra 3S S.P.A.*
Via Giuseppe Valmarana, 63
00139 Rome, Italy
Tel: (06) 81 27 290 - 81 27 324
TELEX: 612051

JAPAN
Asahi Electronics Co. Ltd.
KMM Bldg. Room 407
2-14-1 Asano, Kokura
Kita-Ku, Kiokiyushu City 802
Tel: (093) 511-8471
TELEX: AECOKY 7126-16
Hamilton-Annet Electronics Japan
LTD
YU and YDU Bldg. 1-4 Horidome-Cho
Nihonbashi
Tel: (03) 682-9911
TELEX: 2523774
Nippon Micro Computer Co. Ltd.
Mutsumi Bldg. 4-5-21 Kojimachi
Chiyoda-ku, Tokyo 102
Tel: (03) 230-0041
Ryoyo Electric Corp.
Korwa Bldg.
1-12-22, Tsukiji, 1-Chome
Chuo-Ku, Tokyo 104
Tel: (03) 543-7711
Tokyo Electron Labs. Inc.
No. 1 Hushikata-Machi
Midori-Ku, Yokohama 226
Tel: (045) 471-8811
TELEX: 781-4773

KOREA
Koraim Digital
Room 411 Ahil Bldg.
49-4 2-GA Hoehyun-Dong
Chung-Ku Seoul
Tel: 23-8123
TELEX: HANSINT K23542
Leewood International, Inc.
C.P.O. Box 4046
112-25, Sokong-Dong
Chung-Ku, Seoul 100
Tel: 28-5927
CABLE: "LEEWOOD" Seoul

MEXICO
Provedora Electronica S.A.
(PROESA)
Proi. Mociezuma Ote. 24
Col. Romero de Tereiros
Apdo. Postal 21-139
Mexico 21, D.F.
Tel: 554-8300
TELEX: 017-72402 SAULME

NETHERLANDS
Intelco Nether Comp. Sys. BV
Turflokkerslaan 63
Aalsmeer 1431 GD
Tel: (2977) 28855
TELEX: 14693
Koning & Harlman
Kopervierf 30
2544 EN Den Haag
Holland
Tel: (70) 210 101
TELEX: 31528

NEW ZEALAND
W. K. McLean Ltd
103-5 Feilon Mathew Avenue
Glenn Innes, Auckland, 6
Tel: 587-037
TELEX: NZ2763 KOSFY

NORWAY
Nordisk Elektronik (Norge) A/S
Postoffice Box 122
Smedsvingen 4
1364 Hvalstad
Tel: 02 78 82 10
TELEX: 17546

PORTUGAL
Dlram
Componentes E Electronica LDA
Av. Miguel Bombarda, 133
Lisboa 1
Tel: (19) 545313
TELEX: 14347
SINGAPORE
General Engineers Associates
Bik 3, 1003-1008, 10th Floor
P.S.A. Multi-Storey Complex
Telok Blangah/Pasir Panjang
Singapore 5
Tel: 271-3163
TELEX: RS23987 GENERCO

SOUTH AFRICA
Electronic Building Elements
Pine Square
18th Street
Hazelwood, Pretoria
Tel: (12) 789 221
TELEX: 30181 SA

SPAIN
Interface*
Ronda San Pedro 22
Barcelona 10
Tel: 3/301 7851
TELEX: 51508 IFCE E
Interface
Av. Generalísimo 51 9
E-Madrid 16
Tel: 456 3151
Interface
Calle Bailen 9
Appt 7
Bilbao
Tel: 4/415-0893
ITT SESA
Miguel Angel 16
Madrid 10
Tel: (1) 4190957
TELEX: 27707/27461

SWEDEN
Backstrom Gosta
Box 12009
10221 Stockholm
Tel: (08) 541 080
TELEX: 10135
Nordisk Elektronik AB
Box 27301
S-10254 Stockholm
Tel: (08) 635040
TELEX: 10547

SWITZERLAND
Industrie AG
Gemsenstrasse 2
Postchek 80 - 21190
CH-8021 Zurich
Tel: (01) 60 22 30
TELEX: 56768

TAIWAN
Taiwan Automation Co.*
2nd Floor
224 Nanking East Road
Section 3
Taipei
Tel: 771-0940
TELEX: 11942 TAIAUTO

UNITED KINGDOM
G.E.C. Semiconductors Ltd.
East Lane
North Wembley
Middlesex HA9 7PP
Tel: (01) 904-9303/908-4111
TELEX: 28817

Jermyn Industries
Vestry Estate
Sevenoaks, Kent
Tel: (0732) 501 44
TELEX: 95142
Rapid Recall, Ltd
6 Soho Mills Ind. Park
Woburn Green
Bucks, England
Tel: (6285) 24961
TELEX: 849439
Sinfon Electronics Ltd.*
Arkwright Road 2
Reading, Berkshire RG2 0LS
Tel: (0734) 85464
TELEX: 847395

VENEZUELA
Componentes y Circuitos
Electronicos TTLCA C.A
Aparado 3223
Caracas 101
Tel: 239-0820
TELEX: 21795 TELETIPOS

*Field Application Location
98-01014-01



INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080
Printed in U.S.A./TP-11/1179/5K BA BL